

18.137 / 4 / 03

TUGAS AKHIR
KS 1701



**STUDI PENERAPAN LOGIKA FUZZY
PADA SISTEM OPERASIONAL PERMESINAN
KAPAL PENYEBERANGAN (FERRY)**

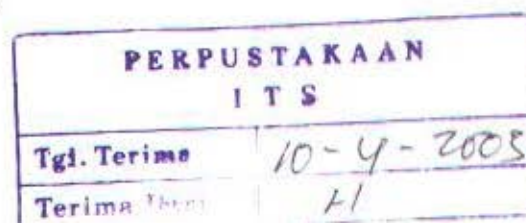


KSSP
629.89
Had
5-1
2003

OLEH :

SOFYAN HADI
NRP. 4298 100 010

**JURUSAN TEKNIK SISTEM PERKAPALAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2003**



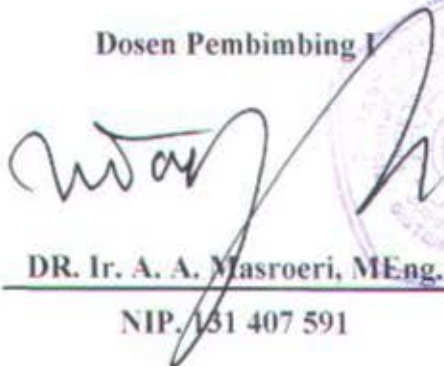
**STUDI PENERAPAN LOGIKA FUZZY PADA
SISTEM OPERASIONAL PERMESINAN KAPAL
PENYEBERANGAN (FERRY)**

TUGAS AKHIR

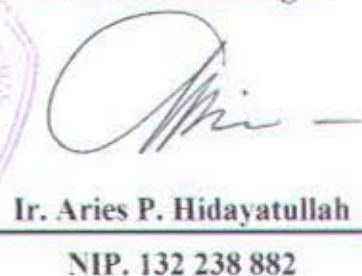
**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana
Pada
Jurusan Teknik Sistem Perkapalan
Fakultas Teknologi Kelautan
Institut Teknologi Sepuluh Nopember
Surabaya**

Mengetahui / Menyetujui

Dosen Pembimbing I


DR. Ir. A. A. Masroeri, MEng.
NIP. 131 407 591

Dosen Pembimbing II


Ir. Aries P. Hidayatullah
NIP. 132 238 882

SURABAYA

JANUARI, 2003



AB STRAK

Sebelum proses starting dilakukan pada motor diesel ada beberapa hal yang harus dilakukan oleh operator-operator kapal di kamar mesin. Secara umum, hal-hal tersebut pada prinsipnya sama meskipun di dunia ini banyak sekali para engine manufacturer yang mengeluarkan produk diesel engine-nya dengan berbagai tipe dan jenis. Proses tersebut sangatlah penting untuk membantu operasional motor induk. Sehingga pada pembahasan di sini dilakukan proses pengaturan proses tahapan-tahapan persiapan sebelum start motor induk. Secara pokok dalam tahapan-tahapan tersebut dua sistem pendukung motor induk yakni sistem pelumas dan sistem pendingin adalah paling penting diperhatikan. Hal ini dikarenakan kedua sistem tersebut harus dijalankan terlebih dahulu untuk menyesuaikan aliran fluida yang dibutuhkan sebelum dilakukan penyalan yakni : oli dan pendinginan air. Namun, dikarenakan masalah kendala waktu penulis hanya membuat program pengaturan yang lebih detail pada sistem pelumas saja sedangkan pada sistem pendingin penulis tidak menjabarkan lebih detail. Pengaturan yang dilakukan di sini dilakukan berdasarkan pengaturan logika fuzzy yang dalam perancangannya tidak memerlukan pendekatan matematis dari plant. Dengan dibuatnya program pengaturan ini diharapkan jikalau diaplikasikan pada kapal, para kru kapal (terutama cadet) bisa dengan mudah memonitor dan lebih cepat untuk beradaptasi dengan semua permesinan yang terkait pada proses tahapan-tahapan tersebut.



KATA PENGANTAR

Dengan mengucapkan Alhamdulillah, puji syukur ke hadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan rangkaian pengerjaan Tugas Akhir ini, mulai dari : pencarian data, pembuatan simulasi sampai pada penulisan Tugas Akhir ini dengan baik.

Hasil dari penulisan Tugas Akhir ini, nantinya diharapkan akan dapat dijadikan salah satu referensi bagi mahasiswa, khususnya mahasiswa Teknik Sistem Perkapalan, serta bagi masyarakat umum untuk menambah wawasan dan pengetahuan tentang proses pengaturan tahapan-tahapan persiapan sebelum dilakukan starting motor induk (khususnya motor disel).

Penulis menyadari sepenuhnya, bahwa penulisan Tugas Akhir ini masih jauh dari sempurna, oleh karena itu, dengan kerendahan hati, penulis menerima saran, kritik dan petunjuk yang membangun untuk kesempurnaan penulisan Tugas Akhir ini dan penulisan-penulisan ilmiah berikutnya. Semoga penulisan Tugas Akhir ini dapat bermanfaat bagi kita semua. Ami.....n, Ami.....n, Ami.....n.

Surabaya, 27 Januari 2003

(Penulis)



DAFTAR ISI

| | |
|---|-------------|
| LEMBAR JUDUL | i |
| LEMBAR PENGESAHAN | ii |
| ABSTRAK | iii |
| KATA PENGANTAR | iv |
| DAFTAR ISI | v |
| DAFTAR TABEL | viii |
| DAFTAR GAMBAR | ix |
| DAFTAR NOTASI | xii |
| | |
| BAB I PENDAHULUAN | 1-1 |
| 1.1 Latar Belakang | 1-1 |
| 1.2 Permasalahan | 1-2 |
| 1.3 Batasan Masalah | 1-3 |
| 1.4 Tujuan Tugas Akhir | 1-4 |
| 1.5 Manfaat Tugas Akhir | 1-5 |
| 1.6 Metodologi | 1-5 |
| 1.7 Sistematika Tugas Akhir | 1-6 |
| | |
| BAB II DASAR TEORI | II-1 |
| 2.1 Prinsip Dasar Himpunan Fuzzy | II-1 |
| 2.1.1 Perbedaan Himpunan Fuzzy Dengan Himpunan Klasik | II-4 |
| 2.1.2 Himpunan Fuzzy | II-7 |
| 2.1.2.1. Defenisi dan Operasi Dasar Himpunan Fuzzy | II-10 |
| 2.1.2.2. Fungsi Keanggotaan Fuzzy | II-12 |
| 2.1.3 Variabel Linguistik | II-13 |



| | | |
|----------------|---|--------------|
| 2.1.4 | Pengatur Logika Fuzzy | II-15 |
| 2.1.4.1. | Struktur Dasar Pengatur Logika Fuzzy | II-17 |
| 2.1.4.2. | Fuzzifikasi | II-18 |
| 2.1.4.3. | Basis Data | II-19 |
| 2.1.4.3.1. | Kuantisasi dan Normalisasi | II-19 |
| 2.1.4.3.2. | Pembagian Ruang Masukan dan Keluaran | II-21 |
| 2.1.4.3.3. | Pemilihan Fungsi Keanggotaan .. | II-22 |
| 2.1.4.4. | Aturan Dasar | II-23 |
| 2.1.4.4.1. | Pemilihan Variabel Masukan dan Keluaran | II-24 |
| 2.1.4.4.2. | Penurunan Aturan Kontrol-kontrol Fuzzy | II-24 |
| 2.1.4.5. | Logika Pengambilan Kesimpulan | II-28 |
| 2.1.4.5.1. | Aturan Kontrol Fuzzy | II-28 |
| 2.1.4.5.2. | Fungsi Implikasi Fuzzy | II-29 |
| 2.1.4.5.3. | Penafsiran Kata Hubung | II-31 |
| 2.1.4.5.4. | Mekanisme Inferensi | II-31 |
| 2.1.4.6. | Defuzzifikasi | II-35 |
| 2.2 | Mekanisme Start Motor Diesel | II-37 |
| 2.2.1 | Kedudukan Menstart | II-40 |
| 2.2.2 | Pemeriksaan Sebelum Start Motor Diesel | II-41 |
| 2.2.3 | Menstart Motor Diesel | II-43 |
| 2.2.4 | Pemanasan Setelah Start Motor Diesel | II-44 |
| BAB III | PEMODELAN SISTEM DAN IMPLEMENTASI LOGIKA FUZZY | III-1 |
| 3.1 | Tinjauan umum | III-1 |
| 3.2 | Model Tahapan-tahapan Persiapan Sebelum Start | III-2 |
| 3.3 | Arsitektur Sistem Pengaturan | III-10 |
| 3.4 | Model Window Interface | III-11 |



| | | |
|-----------------------------|--|-------------|
| 3.5 | Perancangan Pengatur Logika Fuzzy | III-13 |
| 3.5.1 | Penentuan Variabel Masukan dan Keluaran Fuzzy .. | III-14 |
| 3.5.2 | Strategi Pengaturan Pengatur Fuzzy | III-15 |
| 3.5.3 | Basis Data | III-16 |
| 3.5.4 | Penyusunan Aturan Dasar | III-24 |
| 3.5.5 | Logika Pengambilan Keputusan | III-27 |
| 3.5.6 | Defuzzifikasi | III-27 |
| BAB IV | ANALISA DAN PEMBAHASAN | IV-1 |
| 4.1 | Simulasi | IV-1 |
| 4.1.1 | Tampilan Menu Utama | IV-1 |
| 4.1.2 | Tampilan Jikalau Terdapat Trouble | IV-9 |
| 4.2 | Analisa dan Pembahasan Sistem | IV-13 |
| BAB V | PENUTUP | V-1 |
| 5.1 | Kesimpulan | V-1 |
| 5.2 | Saran | V-2 |
| DAFTAR PUSTAKA | | xiii |
| LAMPIRAN | | |



DAFTAR TABEL

| TABEL | HALAMAN |
|---|---------|
| 2.1 Contoh normalisasi himpunan fuzzy pada segmen tertentu .. | II-21 |
| 2.2 Prototipe aturan kontrol linguistik dengan 3 variabel .. | II-26 |
| 2.3 Penyempurnaan aturan kontrol linguistik dengan 3 nilai .. | II-26 |
| 2.4 Penyempurnaan aturan kontrol dengan 7 nilai | II-27 |
| 2.5 Penyempurnaan aturan kontrol dengan 7 nilai | II-27 |



DAFTAR GAMBAR

| GAMBAR | HALAMAN |
|--|---------|
| 2.1 Perbandingan bentuk fungsi keanggotaan μ_F | II-8 |
| 2.2 Derajat keanggotaan himpunan tegas | II-8 |
| 2.3 Derajat keanggotaan himpunan kabur (<i>fuzzy</i>)..... | II-9 |
| 2.4 Bentuk fungsi keanggotaan | II-13 |
| 2.5 Penafsiran grafis variabel linguistik | II-15 |
| 2.6 Blok diagram pengatur logika <i>fuzzy</i> | II-18 |
| 2.7 Penafsiran grafis dari tabel 2.1 yang telah dinormalisasi | II-21 |
| 2.8 Contoh penggunaan fungsi-fungsi keanggotaan <i>fuzzy</i> | II-23 |
| 2.9 Tanggapan sistem loop | II-25 |
| 2.10 Penafsiran grafis lemma 2 | II-34 |
| 2.11 Penafsiran grafis lemma 3 | II-35 |
| 2.12 Hasil defuzzifikasi dengan menggunakan metoda COA atau COG | II-36 |
| 2.13 Sebuah bentuk sistem udara start | II-39 |
| 3.1 Blok diagram tahapan sebelum start motor induk | III-6 |
| 3.2 Blok diagram tahapan pada sistem pelumas | III-7 |
| 3.3 Blok diagram tahapan pada sistem pendinginan | III-8 |
| 3.4 Blok diagram tahapan pada sistem bahan bakar | III-7 |
| 3.5 Blok diagram tahapan pada sistem udara start | III-9 |



| | | |
|------|--|--------|
| 3.6 | Blok diagram tahapan pada sistem udara bilas dan buang .. | III-7 |
| 3.7 | Arsitektur sistem pengaturan | III-10 |
| 3.8 | Model plant pada windows interface | III-12 |
| 3.9 | Fungsi keanggotaan volume oli | III-16 |
| 3.10 | Fungsi keanggotaan lakukan pengisian pada sump tank .. | III-17 |
| 3.11 | Fungsi keanggotaan tekanan hisap pompa oli | III-17 |
| 3.12 | Fungsi keanggotaan tekanan hisap kemasukan udara pompa oli | III-18 |
| 3.13 | Fungsi keanggotaan tekanan keluar pompa oli | III-18 |
| 3.14 | Fungsi keanggotaan delta P3 oli | III-19 |
| 3.15 | Fungsi keanggotaan kondisi tekanan keluar P3 oli | III-19 |
| 3.16 | Fungsi keanggotaan delta P4 oli | III-20 |
| 3.17 | Fungsi keanggotaan kondisi tekanan keluar P4 oli | III-20 |
| 3.18 | Fungsi keanggotaan ampere pompa oli | III-21 |
| 3.19 | Fungsi keanggotaan Bunyi & getaran pompa oli | III-21 |
| 3.20 | Fungsi keanggotaan ampere pompa oli | III-22 |
| 3.21 | Fungsi keanggotaan panas dari pompa oli | III-22 |
| 3.22 | Fungsi keanggotaan temperatur 1 oli | III-23 |
| 3.23 | Fungsi keanggotaan temperatur 2 oli | III-23 |
| 3.24 | Fungsi keanggotaan kondisi kekentalan oli | III-23 |
| 3.25 | Aturan dari kejadian input dan ouput pertama dari sistem oli | III-24 |
| 3.26 | Aturan dari kejadian input dan ouput ketiga dari sistem oli .. | III-25 |
| 3.27 | Aturan dari kejadian input dan ouput keempat dari sistem oli | |
| | (a) untuk ouput bunyi dan getaran dari pompa, (b) untuk panas | |
| | dari pompa | III-25 |



| | |
|--|--------|
| 3.28 Aturan dari kejadian input dan ouput kelima dari sistem oli ... | III-26 |
| 3.29 Aturan dari kejadian input dan ouput ketujuh dari sistem oli... | III-26 |
| 3.30 Aturan dari kejadian input dan ouput keenam dari sistem oli | III-27 |
| 4.1 Tampilan menu utama | IV-1 |
| 4.2 Tampilan window general setting | IV-7 |
| 4.3 Tampilan window setting sistem pelumasan | IV-7 |
| 4.4 Tampilan awal window trouble untuk output panas dengan input ampere pada sistem pelumas | IV-11 |
| 4.5 Tampilan window trouble untuk output panas dengan input ampere pada sistem pelumas | IV-11 |



DAFTAR NOTASI

| | |
|-----------------|--|
| \wedge | Faktor pembobot dengan operasi minimum |
| \cdot | Faktor pembobot dengan produk aljabar dalam produk kartesian |
| μ | Fungsi keanggotaan |
| $\mu_{A,B,C}$ | Fungsi keanggotaan A, B, C |
| $\mu_{\bar{A}}$ | Fungsi keanggotaan komplemen himpunan fuzzy A |
| μ_F | Fungsi keanggotaan F |
| $A \bullet B$ | Perkalian aljabar himpunan A dan B |
| $A \oplus B$ | Penjumlahan aljabar himpunan A dan B |
| e | Error |
| Δe | Perubahan error |
| R_C | Fungsi implikasi aturan operasi mini |
| R_i | Relasi aturan kontrol ke – i |
| u | Sinyal atur |
| Δu | Perubahan sinyal atur |
| \cup | Operasi gabungan (maximum) |
| \cap | Operasi irisan (minimum) |
| \supseteq | Himpunan bagian |
| \in | Elemen |
| Z_o | Defuzzifikasi |



B A B I

PENDAHULUAN

1.1. Latar Belakang

Sebelum proses *starting* dilakukan pada motor diesel ada beberapa hal yang harus dilakukan oleh operator-operator kapal di kamar mesin. Secara umum, hal-hal tersebut pada prinsipnya sama meskipun di dunia ini banyak sekali para *engine manufacturer* yang mengeluarkan produk *diesel engine*-nya dengan berbagai tipe dan jenis. Meskipun juga dari berbagai *diesel engine* tersebut nantinya dipakai oleh jenis kapal yang berbeda-beda sesuai dengan misinya, proses persiapan untuk sebelum dilakukannya *starting* motor secara garis besar adalah tidak jauh berbeda antara satu dengan yang lain. Biasanya yang membuat berbeda adalah salah satunya mengenai pengoperasian sistem start yang digunakan pada kapal tersebut. Apakah memakai sistem start dengan motor listrik, dengan udara tekan atau dengan metoda lainnya.

Mengacu dari sudut pandang tersebut di atas, dibuatlah dalam pembahasan tugas akhir ini untuk mencoba dilakukannya proses pengaturan terhadap beberapa hal yang harus dilakukan oleh si operator kapal di kamar mesin sebelum proses *starting* dijalankan. Beberapa hal tersebut dalam pembahasan di sini disebut sebagai tahapan-tahapan yang akan dilakukan saat motor induk dalam keadaan berhenti sampai saat dilakukannya proses start pada motor tersebut. Sebagai bahan



acuan untuk dibuat studi kasus adalah diambil pada sebuah kapal ferry (penyeberangan) dengan wilayah operasi Ujung – Kamal. Di mana selama ini pada beberapa kapal ferry yang beroperasi di dermaga Ujung – Kamal, untuk melakukan tahapan-tahapan proses sebelum dilakukan starting motor masih dilakukan dengan cara konvensional. Artinya, pekerjaan yang dilakukan oleh si operator tersebut masih memakai jasa dari manusia (biasa disebut sebagai cara manual) tidak digunakan suatu bantuan alat-alat moderen (misalnya, dengan memonitor lewat layar komputer dengan mengontrol beberapa indikator yang ada). Sehingga dari uraian di atas, dibuatlah suatu pengaturan terhadap tahapan-tahapan tersebut di mana dibuat suatu program komputer dengan metoda pengaturan logika *fuzzy*. Dipakainya logika *fuzzy* ini dalam pembahasan di sini adalah dimaksudkan bahwa permasalahan di sini cukup kompleks dan tidak bisa dibuat menjadi suatu persamaan yang linier dan di samping itu logika *fuzzy* dapat diterapkan pada multivariabel dari suatu masalah (*plant*).

1.2. Permasalahan

Hal yang menjadi permasalahan dalam pembahasan tugas akhir ini adalah bagaimana mengoperasikan sistem operasional dari permesinan suatu motor induk di kapal (terutama pada kapal ferry penyeberangan, studi kasus di dermaga penyeberangan Ujung - Kamal), khususnya yang berhubungan dengan persiapan dari motor induk tersebut saat menjalankan proses *starting*. Tentunya dari hal tersebut yakni sebelum motor induk distart, ada beberapa tahapan-tahapan yang harus dilakukan oleh si operator kapal. Dari beberapa tahapan-tahapan yang dilakukan oleh si operator kapal di kamar mesin, selama ini operasionalnya



kebanyakan masih dalam bentuk manual atau tidak dibuat dengan peralatan bantu yang membuat pekerjaan tersebut menjadi lebih praktis. Jadi, dengan kata lain sistem operasional pada hal tersebut masih dilakukan secara konvensional tidak dalam sistem moderen (yakni dengan memonitor lewat layar monitor) dengan sebuah bantuan *software* dan ditambah perangkat elektronik lainnya.

Sehingga, target utama yang ingin dicapai dalam pembahasan Tugas Akhir ini adalah mengaplikasikan logika *fuzzy* dalam bentuk sebuah *software* pada sistem operasional permesinan motor induk di kapal ferry sebagai bahan alternatif, khususnya mengenai hal-hal yang dilakukan saat motor induk dalam keadaan berhenti atau mati sampai motor induk dilakukan proses start. Tentunya sistem operasional yang dilakukan terhadap proses sebelum start motor induk mencakup beberapa sistem pendukung dari motor induk sendiri. Diantara sistem-sistem pendukungnya yaitu : beberapa yang berhubungan dengan sistem pelumasan, sistem pendingin, sistem bahan bakar dan peralatan-peralatan sistem start. Namun kiranya dalam pembahasan di sini tidak semua sistem tersebut dibahas dan dibuat pengaturannya. Artinya hanya diambil sebagian saja seperti yang diuraikan pada batasan masalah.

1.3. Batasan Masalah

Batasan masalah yang diambil antara lain :

1. *Plant* atau proses yang digunakan sebagai objek adalah pada daerah operasi saat motor induk dalam keadaan berhenti sampai dilakukannya proses start.



2. Sistem starting motor induk yang dibahas adalah dengan menggunakan sistem udara bertekanan.
3. Ruang lingkup sistem pendukung motor induk hanya diutamakan pada sistem pelumasan.
4. Tidak membuat alat pengontrol sistem pengatur.
5. Pertimbangan ekonomis tidak dibahas.
6. Bentuk fungsi keanggotaan yang digunakan adalah bentuk segitiga.
7. Sebagai unit defuzzifikasi digunakan metoda *centre of area* atau *centre of gravity*.

1.4. Tujuan Tugas Akhir

Tujuan penulisan tugas akhir ini adalah untuk :

1. Menerapkan metoda kontrol logika fuzzy pada sistem operasional permesinan kapal sebagai metoda pengaturan alternatif, khususnya motor induk di kapal penyeberangan Ujung – Kamal.
2. Membuat rule-rule dari pengaturan logika *fuzzy* yang akan dipakai untuk pengaturan sistem operasional permesinan motor induk yakni tahapan-tahapan yang dilakukan saat motor induk berhenti sampai saat dilakukannya *starting*.



1.5. Manfaat Tugas Akhir

Manfaat yang diharapkan di sini adalah dengan dilakukannya pengontrolan atau pengaturan sistem operasional motor induk (terutama dalam hal persiapan saat motor induk berhenti sampai dilakukannya *starting* hingga *running*) dengan menggunakan logika *fuzzy* diharapkan dapat menghasilkan keuntungan yang lebih besar daripada dilakukannya dengan cara manual. Yakni, dapat membuat cara kerja operator kapal tersebut dapat menjadi lebih mudah dalam menjalankannya walaupun hanya sebagian dari sistem pendukung motor induk.

1.6. Metodologi

Metode-metode yang digunakan untuk mengerjakan tugas akhir ini adalah :

1. Studi literatur

Untuk memperoleh informasi dan masukan tentang segala sesuatu yang mempunyai kaitan dengan judul penelitian.

2. Membuat dasar teori.

3. Mencari data-data yang diperlukan.

4. Memodelkan Sistem dan mengimplementasikan logika *fuzzy* dengan :

- Menentukan variabel linguistik.
- Menentukan Fuzzy Set.
- Menentukan Fuzzy Rules atau algorithm.

5. Pengujian sistem (model *fuzzy*).



6. Menganalisa dan membahas hasil sistem.
7. Membuat kesimpulan dari analisa dan penulisan yang dilakukan pada Tugas Akhir ini.

1.7. Sistematika Tugas Akhir

Abstrak

Kata Pengantar

Daftar Simbol

Daftar Gambar

Daftar Tabel

Daftar Isi

Bab I Pendahuluan

Berisi tentang uraian yang mengenai latar belakang, permasalahan, batasan masalah, tujuan tugas akhir, manfaat tugas akhir, metodologi dan sistematika tugas akhir.

Bab II Dasar Teori

Berisi tentang uraian dasar-dasar teori yang berkenaan dengan prinsip logika fuzzy dan sistem start motor diesel.

Bab III Pemodelan Sistem dan Implementasi Logika *Fuzzy*

Berisi tentang uraian mengenai permasalahan yang akan dibuat menjadi suatu model untuk diterapkan dalam bentuk sebuah simulasi dan



mengimplementasikan logika fuzzy ke dalam model tersebut serta perancangan algoritma (rule-rule) yang digunakan.

Bab IV Analisa dan Pembahasan

Berisi tentang uraian analisa dari perancangan yang dibuat dan pembahasan mengenai performa dari hasil software yang dibuat.

Bab V Kesimpulan dan Saran

Berisi tentang uraian kesimpulan dari seluruh pengerjaan tugas akhir dan pernyataan penulis sebagai saran untuk mengembangkan permasalahan ini lebih lanjut.

Daftar Pustaka

Lampiran



B A B II

DASAR TEORI

2.1. Prinsip Dasar Himpunan Fuzzy

Di era tahun 1285 – 1340, seorang ilmuwan bernama William memberikan pernyataannya mengenai dua nilai logika. Logika tersebut dispekulasikan dalam bentuk “If p Then q”. Di mana kemungkinan jika satu dari dua komponen nilai tersebut dikatakan “benar” atau “salah”. Kemudian pada periode waktu sekitar tahun 1878 – 1956, ilmuwan bernama Lukasiewicz juga memberikan pendapat mengenai nilai logika. Dia memberikan tiga tingkatan nilai logika, yakni : sebagai “benar” (1), “salah” (0) dan “netral” (1/2). Di mana tingkatan “netral” tersebut merepresentasikan “separuh benar” atau “separuh salah”. Dalam kurun waktu itu juga ada beberapa *logician* yang mengembangkan lebih lanjut mengenai *multi level logic*, salah satunya termasuk Lotfi A. Zadeh. (Jamshidi, 1993)

Teori *fuzzy* diprakarsai pertama kali oleh Lotfi A. Zadeh pada tahun 1965 dengan papernya yang dikenal dengan “*Fuzzy set*”. Sebelum mengeluarkan teori *fuzzy*, Zadeh dikenal sebagai seorang sarjana ilmuwan dalam teori kontrol. Dia mengembangkan konsep dari suatu “*state*”, yang dibentuk atas dasar dari teori kontrol moderen. Di awal era 60-an, dia menyatakan pikirannya bahwa teori klasik terlalu banyak menanamkan penekanan pada kepresisian dan oleh karena itu tidak dapat menangani sebuah sistem yang kompleks. Sekitar tahun 1962, dia menulis pernyataannya bahwa untuk menangani sistem biologis, “*Kita*



memerlukan sesuatu yang agak berbeda sama sekali dari matematika, matematika dari fuzzy atau kuantitas dari kabur tersebut adalah tidak dapat dilukiskan dalam istilah suatu distribusi kemungkinan". Sehingga akhirnya, dia memformalkan idenya tersebut dalam bentuk sebuah paper "*Fuzzy Set*".

Di dalam papernya itu, Zadeh memperluas ruang lingkup suatu teori kemungkinan ke dalam sebuah sistem formal dari logika matematis dan memperkenalkan sebuah konsep baru untuk mempergunakan istilah-istilah bahasa secara natural Logika baru ini untuk menggambarkan dan memanipulasikan suatu istilah *fuzzy* (kabur) sehingga dikenal sebagai *fuzzy logic*. Atas hal ini Zadeh dijadikan sebagai "*Master of fuzzy logic*". (Negnevitsky)

Pada dasarnya manusia mengenal objek dengan memberikan klasifikasi secara kualitatif seperti besar, kecil, tinggi, rendah dan sebagainya, yang diklasifikasikan sebagai terminologi linguistik. Batas antara satu kebenaran dengan kebenaran lainnya tidak tegas dan sering mengandung unsur ketidakpastian. Kebenaran ini disebut sebagai kebenaran *fuzzy*.

Seperti contoh ini misalnya, dengan mengatakan *langit biru* bukan berarti bahwa tidak ada sedikitpun *awan* di langit. Akan tetapi kitapun tidak bisa mengatakan bahwa *langit biru* pada saat *langit tertutup awan* lebih dari 50 persen. Adalah wajar jika dikatakan bahwa *langit biru* pada saat *awan* menutupi langit sekitar 10 atau 20 persen. Masalahnya adalah bagaimana batas pemisah harus ditarik. Jika batas pemisah tersebut berada pada 25 persen, apakah berarti 26



persen sudah *tidak lagi biru* ?. Jelas hal ini tidak bisa diterima, sebab nilai 1 persen hampir tidak dapat membedakan karakteristik keadaan *langit biru* dan *tidak biru*. Kita dapat menambahkan kualifikasi bahwa satu persen perbedaan (kurang atau lebih) masih dipandang sebagai *biru*. Akan tetapi, definisi ini pada akhirnya akan membawa kita untuk menerima semua tingkatan *jumlah awan* sebagai *biru*, meskipun *langit mendung*. Untuk mengatasi kontradiksi ini, istilah *biru* dapat disertai oleh *vagueness* (ketidakjelasan) dengan memberikan perubahan transisi bertahap pada derajat *jumlah awan* dari keadaan *biru* menjadi keadaan *tidak biru*. Hal inilah yang menjadi dasar timbulnya konsep himpunan *fuzzy*, yang dicetuskan oleh L. A. Zadeh.

Dapat dikatakan bahwa logika *fuzzy* timbul didasarkan atas premis bahwa nilai logika mutlak tidak mampu memenuhi kebutuhan pemodelan matematis yang ada dalam dunia riil. Bila pada mulanya nilai atas sesuatu lebih didasarkan atas adanya suatu kemutlakan maka dalam logika *fuzzy* mencoba menjembatani antara nilai yang bersifat mutlak tadi sehingga nilai dapat didekati secara lebih riil walaupun dapat membuat adanya suatu *ke-fuzzy-an* dari nilai tersebut. Jembatan tersebut dibangun dengan memakai suatu parameter tambahan yang bertugas untuk menunjukkan seberapa jauh nilai keanggotaan suatu nilai terhadap semestanya dan parameter tersebut menghilangkan kemutlakan atas suatu nilai. Secara matematis dapat dituliskan sebagai berikut :

$$\mu_A(x) = \begin{cases} 1 & \text{jika dan hanya jika } x \in X \\ 0 & \text{jika dan hanya jika } x \notin X \end{cases} \quad (2-1)$$

di mana μ_A merupakan derajat keanggotaan elemen X terhadap semesta A.



Berbasis pengertian di atas maka sebuah keadaan tidak ditampilkan dalam sebuah keadaan tunggal, tetapi diwujudkan dalam sebuah himpunan yang dianggap mampu mewakili nilai tersebut, yakni himpunan *fuzzy* :

$$A = \{x, \mu_A(x), x \in X\} \quad (2-2)$$

Bila X merupakan sebuah himpunan berhingga maka himpunan *fuzzy* dapat dinyatakan sebagai :

$$A = \{(x^1, \mu_A(x^1)), (x^2, \mu_A(x^2)), (x^3, \mu_A(x^3)), \dots, (x^n, \mu_A(x^n))\}$$
$$A = \sum x^i, \mu_A(x^i) \quad (2-3)$$

Tetapi tidak tertutup pula kemungkinan himpunan *fuzzy* sebagai sebuah himpunan tunggal (*singleton fuzzy*). (Budiyanto, 1995)

Algoritma kontrol berbasis teori himpunan *fuzzy* merupakan topik yang telah banyak memberikan hasil selama tahun-tahun terakhir ini. Karakteristik yang paling menarik dari sistem kontrol *fuzzy* ini adalah kemampuannya untuk mengemulsi perilaku manusia (sebagai operator), yang lebih mendasarkan pada kriteria kualitatif daripada yang kuantitatif, dan tidak memerlukan model matematis sistem yang rumit. (Chien Lee, 1990)

2.1.1. Perbedaan Himpunan Fuzzy Dengan Himpunan Klasik

Himpunan klasik (*crisp set*) mempunyai anggota yang sudah pasti atau sekumpulan anggota yang dapat dinyatakan dengan jelas. Hal itu berarti harus



mempunyai kesamaan ciri yang kuat di antara anggota-anggotanya. Ciri ini bersifat tegas sehingga sebuah elemen yang tidak dikenal selalu dapat dinyatakan sebagai di dalam atau di luar suatu himpunan, atau dikenal sebagai logika salah atau benar yang dinyatakan dengan angka nol dan satu. Himpunan klasik memisahkan semesta pembicaraan (*set universal*) menjadi dua bagian, yaitu anggota dan bukan anggota himpunan. Anggota himpunan adalah anggota dari sebuah himpunan klasik yang terdapat pada semesta pembicaraan, sedangkan bukan anggota himpunan berarti anggota semesta pembicaraan yang bukan merupakan anggota himpunan tersebut. Di antara anggota dan bukan anggota himpunan terdapat batas yang jelas atau tajam. Sebuah contoh himpunan klasik berikut ini : suatu himpunan bilangan riil yang mendekati nol, maka dapat dirumuskan sebagai $-0,5 < x < +0,5$. Dengan demikian cukup jelas bahwa batasnya berada pada $-0,5$ dan $+0,5$ sehingga bilangan riil $-0,5$ atau $+0,5$ bukan anggota himpunan bilangan riil yang mendekati nol.

Kenyataan dalam kehidupan sehari-hari banyak pengelompokan atau himpunan yang menggunakan bahasa natural, seperti *kota besar*, *baju mahal*, *gadis cantik*, *langit biru* dan sebagainya, tidak memiliki karakteristik himpunan pasti atau klasik. Batas antara anggota-anggota tersebut terlihat kabur, dan berubah secara bertahap dari anggota himpunan ke bukan anggota himpunan. Dapat dimaksudkan dengan kata lain, suatu yang samar tetapi lebih mudah dimengerti, karena logika di belakang pemikiran manusia adalah logika dengan kebenaran samar (*fuzzy*). Artinya, dapat dilukiskan pada contoh berikut ini : jika manusia mengangkat sebuah balok, secara otomatis di dalam pikirannya *tidak*



terdapat *sekumpulan angka pasti* yang menyatakan berat balok itu. Tetapi manusia akan mengacu pada suatu himpunan samar yaitu **berat balok**. Kemudian manusia mengatakan sebagai “*sangat berat*”, “*cukup berat*”, atau “*agak berat*”. Kata keterangan yang mendahului berat, mewakili tingkat anggota suatu elemen pada himpunan samar tentang berat balok. Dengan dasar itulah disusun suatu teori himpunan samar (*fuzzy set*) sebagai konsep yang lebih umum dari teori himpunan klasik.

Himpunan samar (*fuzzy*) memperkenalkan kekaburan dengan menghapus batas tajam yang memisahkan anggota sebuah himpunan dan bukan anggota himpunan. Sebuah himpunan *fuzzy* dapat didefinisikan secara matematis, dengan memberikan kepada setiap elemen, dalam semesta pembicaraan sebuah nilai yang mewakili keanggotaannya pada himpunan. Nilai keanggotaan tersebut berhubungan dengan tingkat kemiripan atau kesesuaian elemen tersebut dengan konsep yang diwakili oleh himpunan itu. Dengan demikian elemen-elemen pada semesta pembicaraan akan termasuk ke dalam himpunan dengan tingkat kemiripan atau kesesuaian yang lebih besar atau lebih kecil, sesuai dengan nilai keanggotaannya. Nilai-nilai keanggotaannya tersebut biasanya diwakili bilangan riil pada selang tertutup antara 0 sampai dengan 1.

Seperti contoh sebelumnya, sebuah himpunan *fuzzy* yang mewakili konsep *biru* akan memberikan nilai 1 untuk keadaan penutupan sinar matahari oleh awan 0 persen, 0,8 untuk penutupan 20 persen, 0,4 untuk penutupan 30 persen dan 0 untuk penutupan 80 persen. Untuk keanggotaan penuh (*full membership*) diberi



nilai keanggotaan 0 (nol), dengan demikian dapat dikatakan bahwa himpunan klasik adalah kasus khusus dari himpunan *fuzzy*. Karena himpunan klasik hanya mempunyai dua nilai keanggotaan (1 untuk anggota dan 0 untuk bukan anggota) yang diperbolehkan.

Fungsi karakteristik sebuah himpunan *fuzzy* memberikan nilai yang berbeda pada selang tertentu (biasanya pada selang tertutup dari 0 sampai dengan 1) kepada setiap elemen pada semesta pembicaraan dan menandakan tingkat keanggotaan elemen pada himpunan yang ditinjau. Nilai yang lebih besar menyatakan tingkat keanggotaan yang lebih tinggi. Fungsi dengan sifat yang lebih umum ini disebut sebagai fungsi keanggotaan (*membership function*). (Tanujaya, 1999)

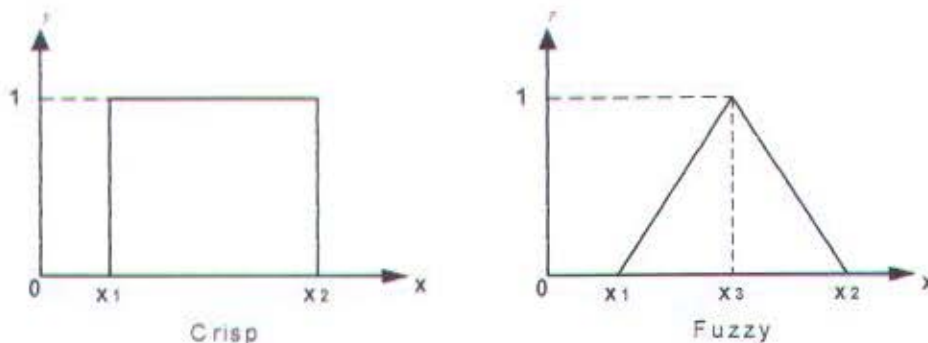
2.1.2. Himpunan Fuzzy

Suatu himpunan *fuzzy* terdiri atas elemen-elemen yang mempunyai derajat keanggotaan μ . Suatu elemen x dalam himpunan bagian *fuzzy* F mempunyai derajat keanggotaan $\mu_F(x)$ yang terletak antara 0 dan 1. Jika $\mu_F(x) = 1$, maka x adalah anggota himpunan F . Jika $\mu_F(x) = 0$, maka x bukan anggota F . Jika $\mu_F(x) = \mu$, dengan $0 < \mu < 1$ maka dikatakan anggota F dengan derajat keanggotaan μ . Dalam merujuk himpunan bagian *fuzzy* sering digunakan besaran-besaran non-numerik, misalkan : *besar, medium, kecil, tinggi, rendah* dan sebagainya. Sebuah contoh, yakni suatu elemen x selain memiliki derajat keanggotaan terhadap himpunan bagian *fuzzy* A juga bisa memiliki derajat keanggotaan terhadap himpunan bagian *fuzzy* lain (misalkan saja B). Derajat keanggotaan suatu elemen



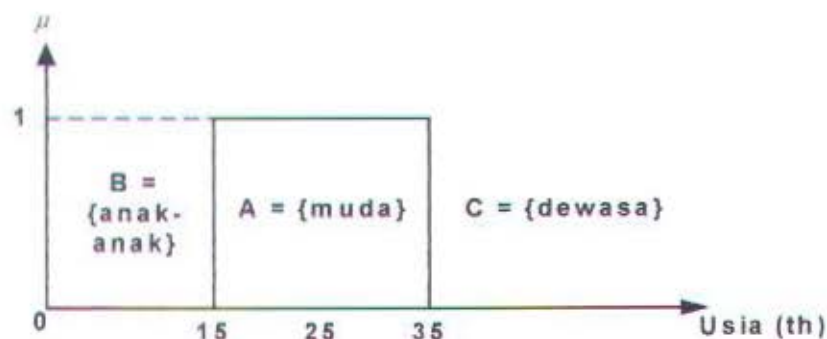
x dalam suatu himpunan bagian *fuzzy* dihitung melalui fungsi distribusi, yang bisa didefinisikan menurut keperluan. Jenis-jenis fungsi distribusi yang populer adalah *segitiga*, *trapesium* dan *eksponensial*.

Bentuk fungsi keanggotaan dari himpunan *fuzzy* dapat berupa segitiga, trapesium, eksponensial dan lain sebagainya. Perbandingan bentuk fungsi keanggotaan himpunan *crisp* dan *fuzzy* dapat diperlihatkan pada gambar 2.1 :



Gambar 2.1. Perbandingan bentuk fungsi keanggotaan F

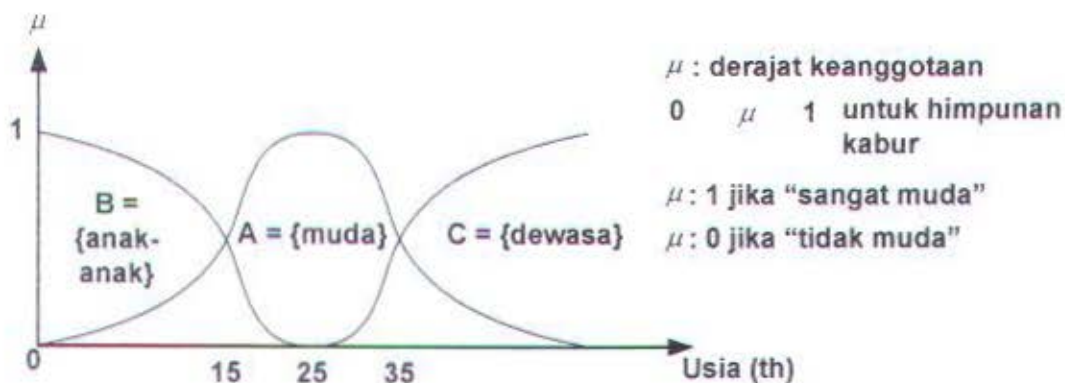
Di bawah ini akan dijelaskan pula suatu contoh untuk menunjukkan tingkat (derajat) keanggotaan baik pada himpunan tegas (*crisp*) maupun pada himpunan *fuzzy*. Contoh tersebut mengenai masalah derajat keanggotaan kelompok-kelompok umur dari sejumlah orang di mana variabel linguistiknya menggunakan 3 variabel yaitu terdiri atas : anak-anak, muda dan dewasa.



Gambar 2.2. Derajat keanggotaan himpunan tegas



Sedangkan jika digunakan derajat keanggotaan berdasarkan himpunan *fuzzy* secara grafis dapat diperlihatkan pada gambar 2.3 di mana pada himpunan *fuzzy* derajat keanggotaannya antara 0 dan 1. Orang usia muda yang didefinisikan $15 \leq x \leq 35$ dalam himpunan tegas, dibagi menjadi tidak begitu muda, agak muda dan sangat muda. Yang tergolong sangat muda mempunyai derajat keanggotaan penuh, yaitu 1. Sedang yang agak muda masih terbagi lagi menjadi agak dewasa dan anak-anak, dengan derajat keanggotaan yang kurang dari 1 ($0 \leq \mu < 1$). Jadi usia 16 dan 17 tahun dalam himpunan kabur dikategorikan sebagai “orang muda” yang derajat kemudaannya sedikit



Gambar 2.3 Derajat keanggotaan himpunan kabur (*fuzzy*)

Dengan gambaran di atas dapat dimengerti bahwa himpunan tegas (*crisp set*) mempunyai keanggotaan yang tegas ($\mu = 1$), sedang himpunan kabur (*fuzzy set*) keanggotaannya bertingkat tergantung masing-masing kategori. (Suharini, 1998)



2.1.2.1. Definisi Dan Operasi Dasar Himpunan Fuzzy

Suatu himpunan *fuzzy* F dalam elemen x didefinisikan sebagai kumpulan pasangan elemen x dan fungsi keanggotaan $\mu_F(x)$. Fungsi keanggotaan $\mu_F(x)$ mempunyai nilai dalam interval $[0,1]$ pada tiap x dalam X .

Nilai fungsi keanggotaan menunjukkan tingkat keanggotaan elemen x dalam F , bila $\mu_F(x) = 1$ menunjukkan x merupakan anggota penuh F , sedang $\mu_F(x) = 0$ menunjukkan x bukan anggota F . Secara umum himpunan *fuzzy* dinotasikan sebagai :

$$F = \{x, \mu_F(x) \mid x \in X\} \quad (2-4)$$

Bila x diskrit dengan n elemen, F dapat dinyatakan sebagai berikut :

$$F = \mu_1(x)/x_1 + \mu_2(x)/x_2 + \dots + \mu_n(x)/x_n$$
$$F = \sum_{i=1}^n \mu_F(x_i)/x_i \quad (2-5)$$

Bila x kontinyu, F dapat dinyatakan sebagai :

$$F = \int_x \mu_F(x)/x \quad (2-6)$$

Dalam himpunan *fuzzy* F , x disebut pendukung (*support*) F . Untuk elemen x di mana $\mu_F(x) = 0,5$ disebut titik silang (*cross-over*). Himpunan *fuzzy* yang mempunyai pendukung tunggal dengan $\mu_F(x) = 1$, disebut *fuzzy* tunggal. Misalkan A dan B adalah himpunan *fuzzy* dalam semesta X dengan fungsi keanggotaan $\mu_A(x)$ dan $\mu_B(x)$. Beberapa operasi dasar dari himpunan *fuzzy* yang sering digunakan adalah sebagai berikut :

**a) Gabungan Fuzzy (Union)**

Jika C adalah gabungan dari A dan B atau $C = A \cup B$, masing-masing mempunyai fungsi keanggotaan $\mu_A(x)$ dan $\mu_B(x)$, maka fungsi keanggotaan himpunan *fuzzy* C adalah :

$$\mu_C(x) = \max \{ \mu_A(x), \mu_B(x) \} \quad (2-7)$$

atau

$$\mu_C(x) = \mu_A(x) \cup \mu_B(x) \quad (2-8)$$

b) Irisan Fuzzy (Intersection)

Jika C adalah irisan dari A dan B atau $C = A \cap B$, yang masing-masing mempunyai fungsi keanggotaan himpunan *fuzzy* C adalah :

$$\mu_C(x) = \min \{ \mu_A(x), \mu_B(x) \} \quad (2-9)$$

atau

$$\mu_C(x) = \mu_A(x) \cap \mu_B(x) \quad (2-10)$$

c) Komplemen Fuzzy

Fungsi keanggotaan dari komplemen himpunan *fuzzy* A adalah sebagai berikut :

$$\mu_{A^c}(x) = 1 - \mu_A(x) \quad (2-11)$$

d) Perkalian Aljabar

Perkalian aljabar himpunan A dan B atau $A \cdot B$ adalah sebagai berikut :



$$\mu_{AB}(x) = \mu_A(x) \sqcap \mu_B(x), \quad x \in X \quad (2-12)$$

e) Penjumlahan Aljabar

Penjumlahan aljabar himpunan A dan B atau $A \oplus B$ adalah sebagai berikut :

$$\mu_{A \oplus B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \sqcap \mu_B(x), \quad x \in X \quad (2-13)$$

2.1.2.2. Fungsi Keanggotaan Fuzzy

Dalam banyak hal fungsi keanggotaan *fuzzy* sangat membantu dan perlu untuk menyatakan fungsi keanggotaan *fuzzy* dalam bentuk standard. Untuk menyatakan suatu fungsi keanggotaan dalam pendefinisian himpunan *fuzzy* tergantung pada bentuk yang akan digunakan. Secara umum ada 2 metode, yaitu pendefinisian secara numerik dan bentuk fungsi.

➤ Definisi secara numerik

Fungsi keanggotaan yang didefinisikan secara numerik menggunakan pendukung diskrit, selain itu dapat juga menggunakan nilai keanggotaan dari bentuk fungsi yaitu dengan mengambil nilai bentuk fungsi untuk tiap pendukung x yang berhingga jumlahnya.

Contoh : $x = \{8, 9, 10, 11, 12\}$

$\mu_A(x) = \{0,5 ; 0,8 ; 1,0 ; 0,8 ; 0,5\}$, maka himpunan fuzzy didefinisikan untuk himpunan elemen x di atas adalah :

$$A = \{0,5/8 ; 0,8/9 ; 1,0/10 ; 0,8/11 ; 0,5/12\}$$



➤ Definisi bentuk fungsi

Fungsi keanggotaan yang didefinisikan dalam bentuk fungsi digunakan pada pendukung kontinyu. Ada beberapa fungsi yang dapat digunakan dalam himpunan fuzzy, bentuk fungsi yang sering digunakan adalah :

✱ Fungsi Eksponensial

$$\mu_A(x) = \exp\left\{-\frac{(x-u)}{2\sigma^2}\right\} \quad (2-14)$$

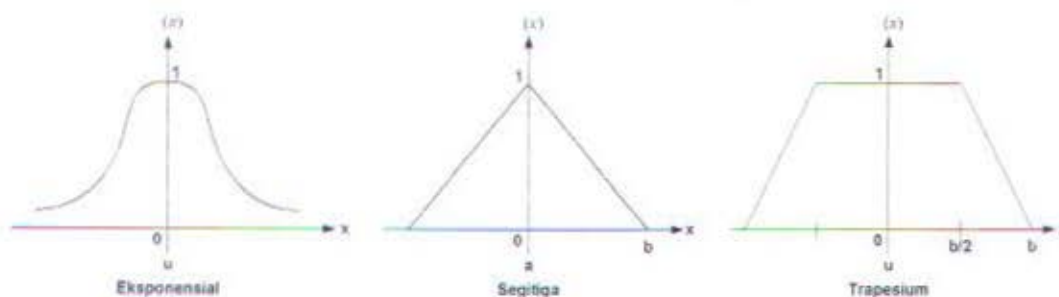
✱ Fungsi Segitiga

$$\mu_A(x) = 1 - \sqrt{\frac{(x-a)^2}{b}} \quad (2-15)$$

✱ Fungsi Trapesium

$$\mu_A(x) = \begin{cases} 1 : 0 \leq (x-a) \leq \frac{b}{2} \\ 2 - 2\sqrt{\frac{(x-a)^2}{b}} : \frac{b}{2} < (x-a) \leq b \end{cases} \quad (2-16)$$

Gambar dari bentuk fungsi di atas ditunjukkan sebagai berikut :



Gambar 2.4 Bentuk fungsi keanggotaan

2.1.3. Variabel Linguistik

Pendekatan dengan himpunan fuzzy untuk merepresentasikan cara berpikir manusia memerlukan variabel linguistik sebagai pengganti dari variabel numerik



yang biasa digunakan dalam pendekatan kuantitatif. Variabel linguistik ini berupa ungkapan-ungkapan yang bersifat kualitatif dan berfungsi untuk menyatakan himpunan *fuzzy*. Dalam berpikir, manusia menggunakan ungkapan yang bersifat kualitatif terhadap informasi-informasi yang diinderanya. Ungkapan-ungkapan kualitatif ini terus menyertai proses berpikir manusia sampai diperoleh pengambilan keputusan yang diungkapkan secara kualitatif juga.

Dalam bentuk sederhana, variabel linguistik dapat dinyatakan dengan pasangan $(u, T(u), X)$. Di mana u menunjukkan nama variabel dan $T(u)$ merupakan istilah yang menyatakan himpunan a , yaitu seperangkat nama-nama nilai linguistik himpunan *fuzzy* atas semesta X . Sebagai contoh, misalkan variabel level akan dinyatakan dengan variabel linguistik, maka seperangkat istilah $T(\text{level})$ dapat dituliskan :

$$T(\text{level}) = \{\text{rendah, sedang}\}$$

Misalkan variabel linguistik didefinisikan untuk semesta $X = [0, 150]$. Dalam semesta ini, rendah terletak di sekitar 50 mm dan sedang di sekitar 100 mm, bila fungsi segitiga digunakan untuk mendefinisikan secara fungsional kedua himpunan ini maka cara menyatakannya adalah :

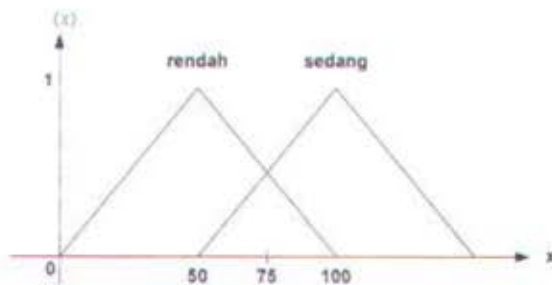
$$\text{Rendah : } \mu_r(x) = 1 - \frac{\sqrt{(x-50)^2}}{b}$$

$$\text{Sedang : } \mu_s(x) = 1 - \frac{\sqrt{(x-100)^2}}{b}$$

di mana $x \in X$. Harga b dipilih sedemikian rupa sehingga titik silang (*cross-over*) terletak di titik $x = 75$ mm, yaitu pendukung dengan nilai keanggotaan $\mu(75) = 0,5$



baik pada himpunan rendah maupun sedang. Penafsiran secara grafis semua pendefinisian tersebut ditunjukkan pada gambar di bawah ini :



Gambar 2.5 Penafsiran grafis variabel linguistik

Gambar tersebut memberikan penafsiran bahwa nilai keanggotaan untuk suatu titik semakin besar bila titik tersebut berada di sekitar titik 50 dan 100. Sedangkan contoh titik 55 mempunyai nilai keanggotaan yang mendekati 1 pada himpunan *rendah*, yang berarti titik ini mempunyai nilai kebenaran yang kuat untuk menjadi anggota himpunan *rendah*, sebaliknya karena mempunyai nilai keanggotaan yang kecil pada himpunan *sedang* maka titik ini mempunyai nilai kebenaran yang lemah untuk menjadi anggota himpunan *sedang*. Sedangkan untuk titik 75 mempunyai nilai kebenaran yang sama untuk menjadi anggota *rendah* maupun *sedang*.

2.1.4. Pengatur Logika Fuzzy

Perancangan Pengatur Logika Fuzzy (*Fuzzy Logic Controller*) adalah menggabungkan aspek pendefinisian himpunan *fuzzy* dengan aspek logika *fuzzy* untuk memperoleh suatu kontroler yang dapat merepresentasikan cara kerja operator manusia. Dengan prosedur perancangan tertentu kedua aspek diterapkan pada masukan dan keluaran untuk membentuk hasil perancangan yang berbentuk suatu algoritma aturan *fuzzy*. Perancangan PLF selama ini tidak mempunyai



prosedur yang baku. Hal ini disebabkan oleh fleksibilitas PLF terhadap berbagai macam *plant* atau proses.

Kickert dan Mamdani memberikan suatu pernyataan yaitu : *"The basic idea behind this aproach was to incoporate the experience of a human process in the design of controller. From a set linguistic rules which describe the operator's control strategy a control algorithm is constructed where the words are defined as a fuzzy set. The main advantages of this aproach seem to be possibility of implementing "rule of thumb", experience, intuition, heuristic, and in the fact that it does not needa model of a process"*.

Ada perbedaan yang mendasar dalam *Fuzzy Logic Control* dengan metoda *Direct digital Control*. Dalam FLC suatu proses diasumsikan bahwa tidak semua parameter tercakup dengan baik dari model atau teori yang sudah tersedia. Berdasar hal tersebut maka FLC mempunyai suatu aturan yang lebih didasarkan atas suatu pengalaman manusia (*human experience*), *rule of thumb* bahkan suatu intuisi. Berdasar hal tersebut maka FLC model adalah *"rule based system"* yang diformulasikan secara eksplisit oleh pendisain FLC system dengan input sebagai observasi dari sistem dan outputnya berupa statemen pengontrol. Cara ini dimaksudkan untuk mempertinggi parameter walaupun dihasilkan suatu ketidakpresisian dari parameter. Untuk itu parameter selalu harus diusahakan agar nilainya selalu tetap terjaga relevansinya. (Budiyanto, 1995)



2.1.4.1. Struktur Dasar Pengatur Logika Fuzzy

Secara umum PLF mempunyai empat bagian pokok seperti yang ditunjukkan oleh Gambar 2.6. Struktur dasar Pengatur Logika Fuzzy mempunyai empat bagian blok, keempat blok tersebut mempunyai fungsi sebagai berikut :

1) Fuzzifier

Berfungsi untuk mentransformasikan sinyal masukan yang bersifat *Crisp* (bukan *fuzzy*) ke himpunan *fuzzy* dengan menggunakan operator *fuzzifier*.

2) Basis Pengetahuan

Berisi basis data dan aturan dasar yang mendefenisikan himpunan *fuzzy* atas daerah-daerah masukan - keluaran dan menyusunnya dalam perangkat aturan kontrol.

3) Logika Pengambil Keputusan

Merupakan inti dari PLF yang mempunyai kemampuan seperti manusia dalam mengambil keputusan. Aksi atur *fuzzy* disimpulkan dengan menggunakan implikasi *fuzzy* mekanisme inferensi *fuzzy*.

4) Defuzzifier

Berfungsi untuk mentransformasikan kesimpulan tentang aksi atur yang bersifat *Crisp* dengan menggunakan operator *defuzzifier*.



Gambar 2.6 Blok Diagram Pengatur Logika Fuzzy
(Patyra, 1996)

2.1.4.2. Fuzzifikasi

Dalam penerapan suatu sistem pengaturan, besaran masukan yang diperoleh dari *plant* melalui sensor akan selalu berupa *Crisp* yang bersifat pasti dan kuantitatif. Sedangkan pengolahan data dalam PLF didasarkan pada teori himpunan *fuzzy* yang menggunakan variabel linguistik yang bersifat *fuzzy*. Oleh karenanya pada tahap awal PLF diperlukan adanya fuzzifikasi yang dilakukan oleh *fuzzifier*. Fuzzifikasi sekaligus dapat dikatakan sebagai pemetaan masukan ke semesta himpunan *fuzzy*. Secara simbolis pemetaan ini dinyatakan dalam $x = \text{fuzzifier}(x_0)$, x_0 adalah masukan *crisp*, x adalah himpunan *fuzzy* dan *fuzzifier* merupakan operator fuzzifikasi.

Metode yang sering dipakai adalah dengan memperlakukan masukan *crisp* yang diperoleh sebagai suatu *fuzzy* tunggal (*fuzzy singleton*). Ini berarti x_0 dianggap sebagai himpunan *fuzzy* dengan fungsi keanggotaan $m(x) = 0$ kecuali pada titik $x_0 = 1$, secara mendasar *fuzzy* tunggal adalah suatu nilai pasti yang



bersifat bukan *fuzzy*. Jadi jelas dalam hal ini tidak ada ke-*fuzzy*-an dalam data yang dihasilkan.

2.1.4.3. Basis Data

Basis data berfungsi untuk mendefinisikan himpunan-himpunan *fuzzy* dari sinyal masukan dan sinyal keluaran agar dapat digunakan oleh variabel linguistik dalam aturan dasar. Perancangan basis data meliputi tiga hal pokok, yaitu :

1. Kuantisasi dan normalisasi.
2. Pembagian ruang masukan dan keluaran.
3. Pemilihan fungsi keanggotaan.

2.1.4.3.1. Kuantisasi dan Normalisasi

Kuantisasi berarti mendiskusikan semesta pembicaraan yang kontinyu ke dalam sejumlah segmen-segmen tertentu yang disebut level kuantisasi. Pemberian nomor atau label level-level ini membentuk pendukung himpunan *fuzzy* secara berhingga atau semesta pembicaraan baru yang bersifat diskrit. Prosedur ini diperlukan bila pendefinisian fungsi keanggotaannya dinyatakan secara numerik. Pendefinisian ini biasanya dinyatakan dalam bentuk tabulasi, yang penerapannya berbentuk suatu tabel pandang (*look-up table*). Tabel tersebut mendefinisikan secara numerik, misalnya dengan pembagian ruang sebanyak 5 variabel linguistik, yaitu : **NB** (Negatif Besar), **NK** (Negatif Kecil), **NL** (Nol), **PK** (Positif Kecil) dan **PB** (Positif Besar). Pengambilan variabel linguistik ini juga tergantung pada si perancangannya. Jadi pengambilannya tidak harus sama seperti di atas.



Pemilihan jumlah level kuantisasi (penyokong) mempengaruhi kepekaan PLF terhadap masukan dan kehalusan aksi atur pada keluaran. Semakin banyak level kuantisasi yang diaplikasikan pada semesta pembicaraan masukan dan keluaran maka semakin peka PLF tersebut dan semakin halus aksi aturnya. Tetapi karena tabel pandang menggunakan sejumlah memori pada komputer maka dalam menentukan jumlah level harus ada kompromi dengan aspek penghematan memori.

Normalisasi diperlukan bila diinginkan semesta pembicaraan yang terbatas pada jangkauan tertentu, misal dari -1 sampai 1 . Jadi normalisasi merupakan pemetaan semesta pembicaraan masukan ke semesta pembicaraan baru yang terbatas. Skala pemetaan bisa seragam (*uniform*) atau tidak seragam (*non-uniform*) tergantung kebutuhan perancangannya. Normalisasi juga diperlukan bila fungsi keanggotaannya didefinisikan secara fungsional. Tabel 2.1 menunjukkan contoh normalisasi dari semesta pembicaraan $[-8, 6]$ ke interval yang sudah dinormalisasi $[-1, 1]$ dan pendefinisianya dengan fungsi segitiga :

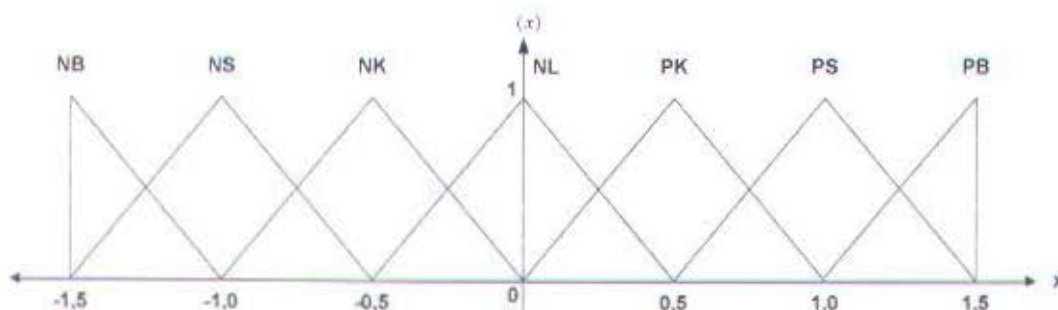
$$\mu(x) = \frac{\sqrt{((x-a)^2)}}{b}$$

Tafsirannya secara grafis diperlihatkan pada Gambar 2.7, yang menunjukkan bahwa normalisasi menghasilkan semesta pembicaraan yang simetris. Jika tidak dinormalisasi maka semesta pembicaraannya akan menjadi asimetris dan pembagiannya tidak merata.



Tabel 2.1 Contoh normalisasi himpunan fuzzy pada segmen tertentu.

| Segmen | Segmen Ternormalisasi | b | u | Himpunan Fuzzy |
|------------|-----------------------|-----|------|----------------|
| $[-8, -4]$ | $[-1,5, -1,0]$ | 0,5 | -1,5 | NB |
| $[-6, -2]$ | $[-1,5, -0,5]$ | 0,5 | -1 | NS |
| $[-4, 0]$ | $[-1,0, 0,0]$ | 0,5 | -0,5 | NK |
| $[-2, 2]$ | $[-0,5, 0,5]$ | 0,5 | 0 | NL |
| $[0, 4]$ | $[0,0, 1,0]$ | 0,5 | 0,5 | PK |
| $[2, 6]$ | $[0,5, 1,5]$ | 0,5 | 1 | PS |
| $[4, 8]$ | $[1,0, 1,5]$ | 0,5 | 1,5 | PB |



Gambar 2.7 Penafsiran grafis dari Tabel 2.1 yang telah dinormalisasi

2.1.4.3.2. Pembagian Ruang Masukan dan Keluaran

Pendefinisian himpunan *fuzzy* atas ruang masukan dan keluaran berarti pula membagi-bagi semesta pembicaraan atas nilai-nilai variabel linguistik seperti : Negatif Besar (NB), Negatif Sedang (NS), Negatif Kecil (NK), Nol (NL), Positif Kecil (PK), Positif Sedang (PS), Positif Besar (PB) dan sebagainya. Pembagian ruang masukan dan keluaran ini menentukan berapa banyak jumlah himpunan ini juga menentukan kehalusan Pengaturan Logika Fuzzy (PLF). Contoh pada Tabel



2.1 dan Gambar 2.7 menunjukkan pembagian ruang dengan 7 himpunan *fuzzy*.
(Chien Lee, 1990)

2.1.4.3.3. Pemilihan Fungsi Keanggotaan

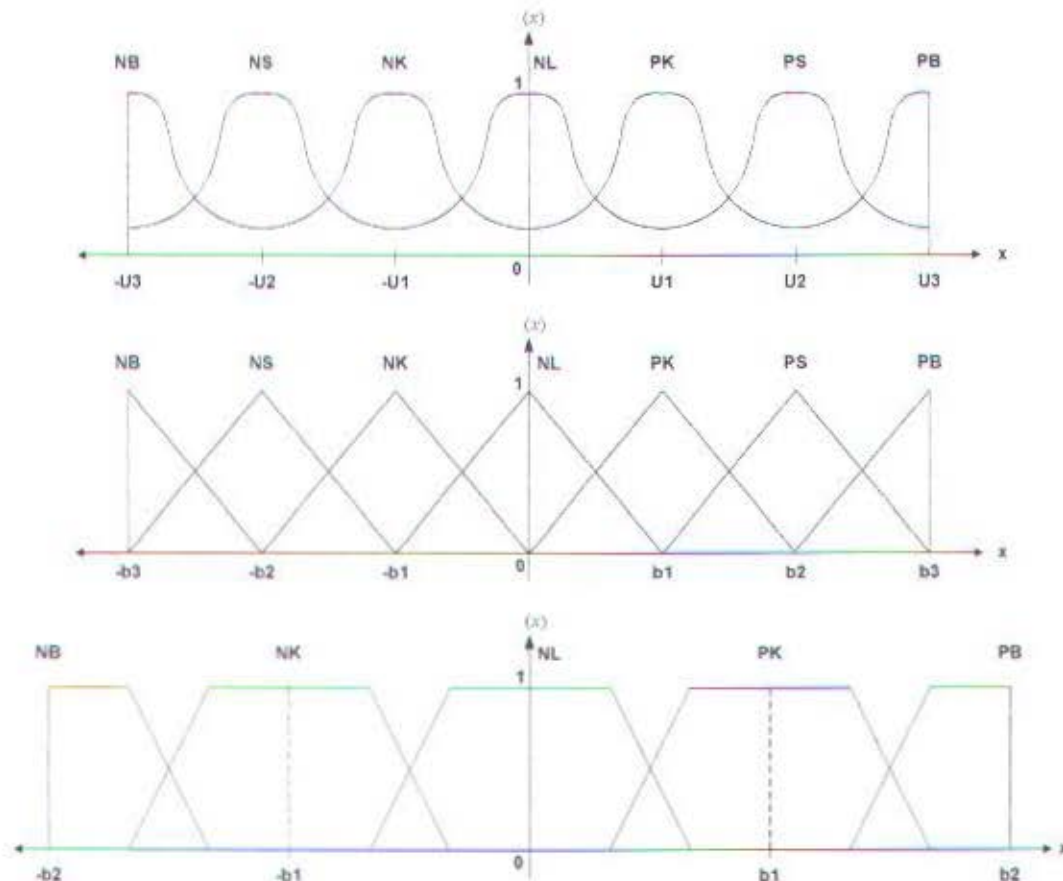
Pada pendefinisian secara numerik tingkat keanggotaan penyokong dalam himpunan *fuzzy* dinyatakan dalam bentuk tabulasi, seperti yang dicontohkan pada Tabel 2.1. Pembagian ruang dilakukan dengan 7 nilai linguistik himpunan *fuzzy*, yaitu : Negatif Besar (NB), Negatif Sedang (NS), Negatif Kecil (NK), Nol (NL), Positif Kecil (PK), Positif Sedang (PS) dan Positif Besar (PB) dengan fungsi keanggotaannya adalah segitiga.

Fungsi keanggotaan dapat dipilih secara bebas dengan menentukan secara sembarang nilai keanggotaannya. Tetapi pada dasarnya penentuan ini harus tetap dapat menggambarkan karakteristik masing-masing himpunan *fuzzy*. Fungsi yang sering digunakan adalah fungsi eksponen (π), fungsi segitiga dan fungsi trapesium dikarenakan fungsi-fungsi tersebut mudah dimanipulasi secara matematis.

Pada pendefinisian secara fungsional tidak diperlukan pemberian nomor level kuantisasi karena dalam hal ini tidak diperlukan penyokong yang diskrit. Pembagian ruang biasanya cukup dengan menentukan titik sumbu simetri dan sebaran (jangkauan) fungsi yang digunakan. Fungsi ini mudah diadopsi terhadap keadaan semesta pembicaraan dengan cara mengatur parameter ratahan yang menentukan titik kerja dan parameter sebaran yang menentukan jangkauan kerja.



Parameter rataaan (u dan a) dan parameter sebaran (b dan s) dapat dipilih secara sembarang, tetapi dengan pertimbangan perancangan yang layak. Dalam hal parameter sebaran seragam biasanya sebaran dipilih sedemikian rupa sehingga titik silang (*cross-over*) terletak di tengah-tengah dua rataaan himpunan yang bersebelahan. Contoh penggunaan fungsi eksponen (π), fungsi segitiga dan fungsi trapesium dapat dilihat pada Gambar 2.8. (Harris, 1993)



Gambar 2.8 Contoh penggunaan fungsi-fungsi keanggotaan fuzzy

2.1.4.4. Aturan Dasar

Sistem *fuzzy* dikarakteristikan oleh istilah-istilah linguistik yang didasarkan pada pengetahuan pakar dan biasanya berbentuk aturan **Jika – Maka**



(*If – Then*) (Yan, 1994), yang disebut aturan kontrol *fuzzy*. Hal ini untuk mempermudah implementasi bentuk kondisional *fuzzy*. Seperangkat aturan kontrol *fuzzy* merupakan penjabaran dari bentuk-bentuk kondisional *fuzzy* dalam aturan dasar pada PLF. Perancangan aturan kontrol meliputi :

1. Pemilihan variabel masukan dan keluaran.
2. Penurunan aturan kontrol *fuzzy*.

2.1.4.4.1. Pemilihan Variabel Masukan dan Keluaran

Pemilihan variabel masukan dan keluaran memberi pengaruh kuat pada karakteristik kontroler. Dalam hal ini pengetahuan dan pengalaman kerekayasaan memainkan peranan yang sangat. Pemilihan variabel ini memerlukan pemahaman tentang perilaku *plant* dan perilaku pengaturnya.

Secara umum sebagaimana kontroler konvensional, *error* dan *perubahan error* dari *plant* tetap menjadi awam utama bagi aturan kontrol PLF untuk menghasilkan sinyal atur. PLF biasanya menggunakan variabel masukan berupa *error* dan perilaku *error* seperti *perubahan error*, turunan *error*, penjumlahan *error*, percepatan *error* dan sebagainya, sedangkan keluaran atau aksi aturnya berupa sinyal atur atau perubahan sinyal atur.

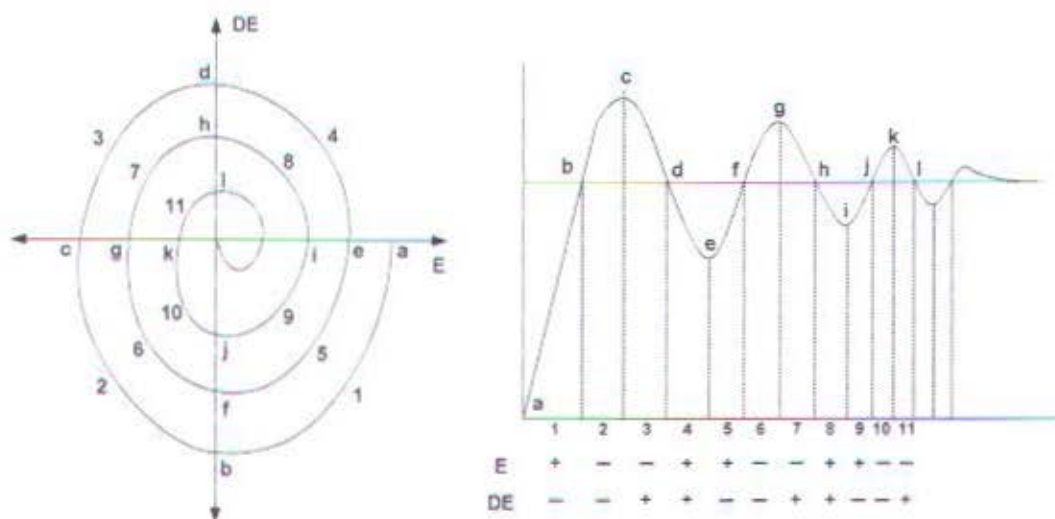
2.1.4.4.2. Penurunan Aturan Kontrol-kontrol Fuzzy

Salah satu cara menurunkan aturan kontrol-kontrol *fuzzy* adalah dengan mengumpulkan aturan-aturan kontrol yang dibentuk dari analisa perilaku objek atur secara heuristik. Aturan kontrol diturunkan dengan jalur mengoreksi



simpangan keluaran *plant* dari keadaan yang diinginkan. Penurunan benar-benar dilakukan dengan secara heuristik dengan mengandalkan pengetahuan kualitatif. Hasil yang diperoleh akan berupa *prototype* aturan kontrol secara garis besar sehingga hasilnya masih perlu disempurnakan.

Ada beberapa metode yang dapat digunakan untuk menyempurnakan *prototype* aturan kontrol itu. King dan Mamdani, mengusulkan metode penyempurnaan aturan kontrol dengan cara yang disebut Pemetaan Skala (*Scale Mappings*). Prinsipnya adalah menentukan sinyal atur sedemikian rupa sehingga trayektori sistem *loop* tertutup terakhir di keadaan yang diinginkan. Dalam hal ini pengetahuan tentang perilaku peralihan *plant*, seperti lewatan, waktu naik dan sebagainya, serta intuisi tentang perilaku sistem tertutup diperlukan.



Gambar 2.9 Tanggapan sistem loop tertutup

Gambar 2.9 menunjukkan tanggapan *plant* loop tertutup terhadap variabel masukan *plant*, yaitu *error* (e) dan perubahan *error* (Δe). Untuk memperjelas trayektori dapat digunakan grafik yang menyatakan hubungan antara *error* (e) dan



perubahan *error* (Δe). Sedangkan variabel keluaran PLF berupa perubahan sinyal atur (Δu) atau sinyal atur (u).

Pada penurunan *prototipe* ini pembagian ruang masukan adalah 3, yang berarti ada 3 nilai linguistik yaitu Positif (P), Negatif (N) dan Nol (NL). Peninjauan kualitatif per titik (bertanda huruf) pada grafik memberikan penalaran yang menghasilkan *prototipe* aturan kontrol pada tabel 2.2.

Tabel 2.2 Prototipe aturan kontrol linguistik dengan 3 nilai

| Aturan No. | e | Δe | u | Titik referensi |
|------------|----|------------|----|-----------------|
| 1 | P | NL | P | a, e, i |
| 2 | NL | N | N | b, f, j |
| 3 | N | NL | N | c, g, k |
| 4 | NL | P | P | d, h, l |
| 5 | NL | NL | NL | set point |

Sedangkan peninjauan secara kualitatif per daerah (bertanda angka) pada grafik tanggapan maupun pada bidang fasa menunjukkan bahwa daerah 1 (satu) mempunyai pengaruh pada waktu pemendekan waktu naik (*rise time*), dan pada daerah 2 (dua) berhubungan dengan pengurangan lewatan (*over shoot*), secara heuristik penalaran ini memberikan penambahan atau penyempurnaan aturan kontrol seperti yang ditunjukkan pada tabel 2.3.

Tabel 2.3 Penyempurnaan aturan kontrol linguistik dengan 3 nilai

| Aturan No. | e | Δe | u | Titik referensi |
|------------|---|------------|----|-----------------|
| 6 | P | N | P | 1 (rise time) |
| 7 | N | N | N | 2 (over shoot) |
| 8 | N | P | N | 3, 7 |
| 9 | P | P | P | 4, 8 |
| 10 | P | N | NL | 9 |
| 11 | N | P | NL | 10 |



Kinerja yang lebih baik dapat diperoleh dengan pembagian ruang yang lebih banyak, misalnya 7 nilai yaitu : NB, NS, NK, NL, PK, PS dan PB. Pembagian ruang dengan 7 nilai sering digunakan pada perancangan PLF, karena pembagian ruang tersebut dapat mewakili ruang keseluruhan dan didapatkan hasil yang baik. *Prototipe* dan penyempurnaan berturut-turut ditunjukkan oleh tabel 2.4 dan 2.5. Secara langsung pembagian ruang masukan mempengaruhi jumlah aturan kontrol yang dihasilkan. Pada sistem **MISO** (*Multi Input Single Output*), jumlah aturan kontrol yang dihasilkan adalah 7×7 aturan kontrol.

Tabel 2.4 Penyempurnaan aturan kontrol dengan 7 nilai

| Aturan No. | e | Δe | u | Titik referensi |
|------------|----|------------|----|-----------------|
| 1 | PB | NL | PB | a |
| 2 | PS | NL | PS | e |
| 3 | PK | NL | PK | i |
| 4 | NL | NB | NB | b |
| 5 | NL | NS | NS | f |
| 6 | NL | NK | NK | j |
| 7 | NB | NL | NB | c |
| 8 | NS | NL | NS | g |
| 9 | NK | NL | NK | k |
| 10 | NL | PB | PB | d |
| 11 | NL | PS | PS | h |
| 12 | NL | PK | PK | l |
| 13 | NL | NL | NL | set point |

Tabel 2.5 Penyempurnaan aturan kontrol dengan 7 nilai

| Aturan No. | e | Δe | u | Titik referensi |
|------------|----|------------|----|-----------------|
| 14 | PB | NK | PS | 1 |
| 15 | PK | NB | NS | 1 |
| 16 | NB | PS | NS | 3 |
| 17 | NS | PB | PS | 3 |
| 18 | PK | NK | NL | 9 |
| 19 | NK | PK | NL | 11 |



2.1.4.5. Logika Pengambilan Kesimpulan

Inti persoalan dari pengaturan *fuzzy* terletak pada logika pengambilan keputusannya yang meniru pengambilan keputusan pada manusia. Untuk memahami proses pengambilan keputusan dalam *fuzzy*, pada bagian ini akan dibahas beberapa konsep logika pengambilan keputusan yang meliputi fungsi-fungsi implikasi *fuzzy*, penafsiran kata hubung, operator komposional dan mekanisme inferensi.

2.1.4.5.1. Aturan Kontrol Fuzzy

Seorang operator yang sedang bekerja secara sadar atau tidak sadar akan menggunakan aturan kontrol yang berupa hubungan **Jika – Maka** dalam mengambil keputusan. Aturan kontrol ini dilengkapi dengan variabel-variabel linguistik sehingga terbentuk suatu pernyataan-pernyataan linguistik yang didasarkan pada pengetahuan dan pengalaman tentang objek atur yang akan diaturnya.

Perilaku suatu sistem pengetahuan *fuzzy* didasari oleh seperangkat pernyataan linguistik berbasis pengetahuan yang meniru cara kerja operator ini. Pernyataan-pernyataan ini menggunakan variabel linguistik dari himpunan *fuzzy*, oleh karenanya disebut pernyataan kondisional *fuzzy* atau aturan kontrol *fuzzy*.

Bentuk umum aturan kontrol *fuzzy* adalah pernyataan **Jika – Maka** (*If – Then*) dan pernyataan lengkapnya adalah (*Yan, 1994*) :



Jika (masukan i) – **Maka** (keluaran j), $i = 1, 2, \dots, n$, $j = 1, 2, \dots, n$ (2 – 17)

Masukan dan keluaran merupakan himpunan *fuzzy* yang dinyatakan dengan variabel linguistik. Masukan merupakan keadaan dari objek atur dan keluaran merupakan tindakan yang harus dilakukan terhadap objek atur sehingga sehubungan dengan keadaan masukan. Dari bentuk aturan kontrol di atas aturan tersebut merupakan bentuk **MIMO (Multi Input Multi Output)**. Untuk sistem **MISO (Multi Input Single Output)**, maka keluarannya hanya ada satu. Sebuah contoh sistem MISO dengan dua masukan dan satu keluaran yang perangkat aturan kontrolnya berbentuk :

$$\begin{array}{ll}
 R_1 & = \text{Jika } x \text{ adalah } A_1 \text{ dan } y \text{ adalah } B_1 \text{ Maka } z \text{ adalah } C_1 \\
 R_2 & = \text{Jika } x \text{ adalah } A_2 \text{ dan } y \text{ adalah } B_2 \text{ Maka } z \text{ adalah } C_2 \\
 & \vdots \\
 R_n & = \text{Jika } x \text{ adalah } A_n \text{ dan } y \text{ adalah } B_n \text{ Maka } z \text{ adalah } C_n
 \end{array}$$

di mana x dan y adalah variabel masukan dan z variabel keluaran. A, B, C adalah nilai linguistik dari variabel x, y dan z dalam semesta pembicaraan X, Y dan Z .

2.1.4.5.2. Fungsi Implikasi Fuzzy

Aturan kontrol pada dasarnya adalah relasi *fuzzy* yang dinyatakan sebagai suatu hubungan sebab akibat yang disebut implikasi *fuzzy*. Aturan kontrol *fuzzy* bila dinyatakan dengan implikasi *fuzzy* akan menjadi :

$$\begin{aligned}
 \mu_{R_i} &= \mu(A_i \text{ dan } B_i) \rightarrow C_i(x, y, z) \\
 &= [\mu_{A_i}(x) \text{ dan } \mu_{B_i}(y)] \rightarrow \mu_{C_i}(z) \quad (2 - 18)
 \end{aligned}$$



A_i dan B_i adalah himpunan *fuzzy* $A_i \times B_i$ dalam $x \times y$, $R_i \subseteq (A_i \text{ dan } B_i) \rightarrow C_i$ adalah implikasi *fuzzy* dalam $X \times Y \times Z$, dan notasi \rightarrow merupakan fungsi implikasi yang pernah diusulkan oleh para peneliti. Namun dari penelitian yang dilakukan oleh Baldwin dan Pilsworth, kemudian oleh Fukami, Mizumoto dan Tanaka diperoleh bahwa hanya beberapa fungsi yang layak dipilih untuk ditetapkan dalam logika pengampilan keputusan pada sistem pengaturan.

Fungsi implikasi dapat didefinisikan sebagai suatu relasi yang menunjukkan suatu keterkaitan antara suatu kejadian sebab dengan kejadian akibat. Semakin cocok hasil pembuktian matematis (secara *fuzzy*) suatu fungsi implikasi dengan intuisi tentang keterkaitan kejadian di atas maka semakin baik implikasi tersebut.

Salah satu fungsi implikasi yang akan dipakai di sini adalah hasil penelitian Mamdani. Di mana fungsi tersebut didasarkan atas operator inferensi Max – Min. Fungsi ini disebut fungsi implikasi aturan operasi mini (*mini-operation rule*) yang dinotasikan dengan R_C :

$$\begin{aligned} R_C &= A \times B \\ &= \int_{x,y} \mu_A(x) \times \mu_B(y) (x,y) \end{aligned} \quad (2-19)$$

jika diterapkan untuk sistem MISO didapatkan :

$$R_C = (A \times B) \times C$$



$$= \int_{x \times y \times z} \mu_{A \times B}(x, y) \wedge \mu_C(z) | (x, y, z) | \quad (2-20)$$

Bentuk $A \times B$ adalah produk kartesian masukan dalam $X \times Y$.

2.1.4.5.3. Penafsiran Kata Hubung

Kata tiap-tiap aturan kontrol *fuzzy* dinyatakan dengan suatu relasi, maka perilaku keseluruhan sistem *fuzzy* ini ditentukan oleh relasi-relasi *fuzzy* ini. Dengan kata lain perilaku sistem *fuzzy* ditentukan oleh relasi tunggal yang merupakan kombinasi seluruh perangkat relasi *fuzzy* dalam sekumpulan aturan kontrol. Pengombinasian ini dilakukan oleh kata hubung. Untuk kata hubung dan (*and*) biasanya digunakan pada produk kartesian yang dilatarbelakangi oleh nilai yang berbeda. Sebagai contoh adalah : “**Jika** (A dan B) – **Maka** C”, sedangkan kata hubung **Juga** (*also*) digunakan pada fungsi implikasi *fuzzy* dan dapat juga sebagai gabungan dan irisan.

2.1.4.5.4. Mekanisme Inferensi

Penyimpulan seperti yang dicontohkan pada operator komposisional diterapkan pada suatu aturan kontrol, sedangkan sistem pengaturan *fuzzy* terdiri dari seperangkat aturan kontrol. Untuk mengombinasikan aturan kontrol tersebut digunakan kata hubung **Juga**. Bentuk umum dari relasi kata hubung **Juga** adalah :

Masukan : x adalah A' dan y adalah B'

R_1 : **Jika** x adalah A_1 dan y adalah B_1 **Maka** z adalah C_1



Juga R_2 : **Jika** x adalah A_2 dan y adalah B_2 **Maka** z adalah C_2

Juga R_n : **Jika** x adalah A_n dan y adalah B_n **Maka** z adalah C_n

Kesimpulan : z adalah C^* (2 – 21)

Keseluruhan aturan kontrol ini harus dapat menghasilkan suatu keluaran berupa kesimpulan yang dinyatakan dengan C^* . Kesimpulan C^* diperoleh dari mekanisme inferensi dengan komposisional sup-star yang didasarkan dari fungsi implikasi *fuzzy* dan kata hubung “**Dan**” dan “**Juga**”.

Ada beberapa Lemma penting yang berhubungan dengan mekanisme inferensi, Lemma-lemma itu adalah :

Lemma 1 : $(A', B') \circ \bigcup_{i=1}^n R_i = (A', B') \circ R_i$ (2 – 22)

R_i adalah relasi aturan kontrol ke – i .

Lemma 2 : Untuk fungsi implikasi-mini Mamdani diperoleh :

$$1. (A', B') \circ (A_i \text{ dan } B_i \rightarrow C_i) = [A' \circ (A_i \rightarrow C_i)] \cap [B' \circ (B_i \rightarrow C_i)] \quad (2 - 23)$$

$$\text{Jika } \mu_{A_i \times B_i} = \mu_{A_i} \wedge \mu_{B_i}$$

$$2. (A', B') \circ (A_i \text{ dan } B_i \rightarrow C_i) = [A' \circ (A_i \rightarrow C_i)] \bullet [B' \circ (B_i \rightarrow C_i)] \quad (2 - 24)$$

$$\text{Jika } \mu_{A_i \times B_i} = \mu_{A_i} \wedge \mu_{B_i}$$

Lemma 3 : Jika masukan berupa *fuzzy* tunggal $A' = X_0$ dan $B' = Y_0$ diperoleh :



$$1. R_C : \alpha_i^{\wedge} \wedge \mu_{C_i}(z), \quad \alpha_i^{\wedge} = \mu_{A_i}(x_o) \wedge \mu_{B_i}(y_o) \quad (2-25)$$

$$2. R_C : \alpha_i^{\bullet} \wedge \mu_{C_i}(z), \quad \alpha_i^{\bullet} = \mu_{A_i}(x_o) \bullet \mu_{B_i}(y_o) \quad (2-26)$$

Lemma 1 menyatakan bahwa keluaran yang disimpulkan dari semua perangkat aturan kontrol sama dengan gabungan semua keluaran tiap-tiap aturan kontrol. Sedang lemma 2 dan 3 menguraikan penerapan fungsi implikasi operasi mini Mamdani (R_o) dengan menggunakan kedua jenis produk kartesian, dari kedua lemma ini diperoleh hubungan sederhana antara masukan dan keluaran :

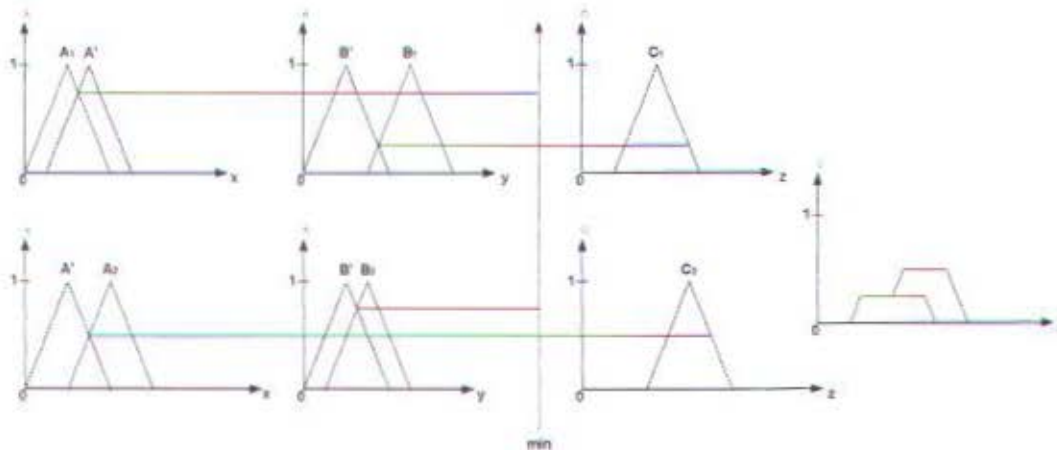
$$R_C : \mu_C = \bigcup_{i=1}^n \alpha_i \wedge \mu_{C_i} \quad (2-27)$$

faktor pembobot α_i , merupakan besarnya kontribusi aturan ke i untuk aksi kontrol *fuzzy*. Faktor pembobot dapat ditentukan dengan dua pilihan yaitu dengan operasi minimum (α_i^{\wedge}) dan produk aljabar (α_i^{\bullet}) dalam produk kartesian. Lemma 2 mensyaratkan masukan bagi aturan kontrol yang berupa suatu himpunan *fuzzy*. Untuk menjelaskan hal itu, diasumsikan ada 2 aturan kontrol *fuzzy* sebagai berikut :

R_1 = Jika x adalah A_1 dan y adalah B_1 maka z adalah C_1

R_2 = Jika x adalah A_2 dan y adalah B_2 maka z adalah C_2

yang mendapat masukan himpunan *fuzzy* A' dan B' . Dan penafiran grafis dari lemma 2 untuk dua aturan kontrol *fuzzy* di atas yang mekanisme inferensinya menggunakan operasi minimum, ditunjukkan pada Gambar 2.10.



Gambar 2.10 Penafsiran grafis Lemma 2

Penafsiran secara grafis Lemma 2 pada Gambar 2.10 menggambarkan bahwa pada mekanisme inferensi, setiap aturan kontrol mempunyai kontribusi terhadap sinyal atur yang diputuskan. Kenyataan bahwa masukan PLF diperlakukan sebagai *fuzzy* tunggal menyebabkan permasalahan dalam mekanisme inferensi lebih sederhana. (Chien Lee, 1990)

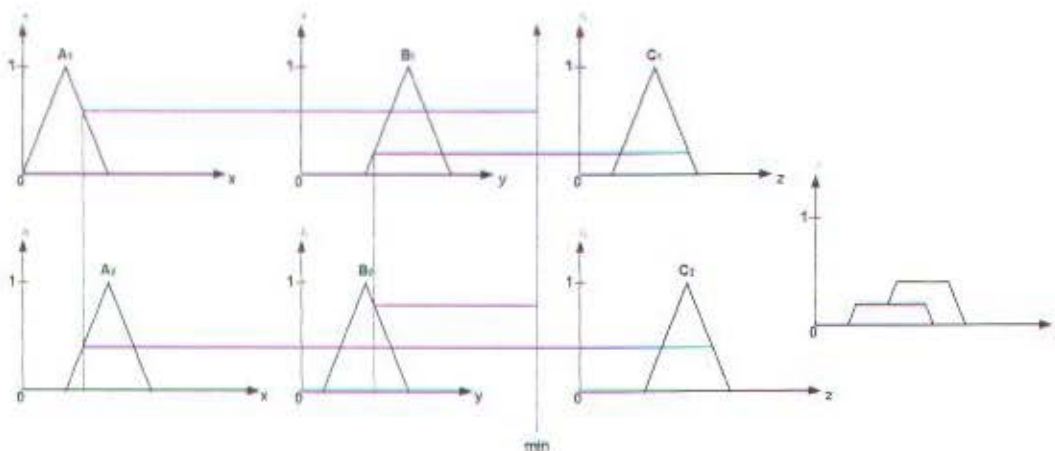
Pada Lemma 3 permasalahan tersebut dapat diatasi. Lemma 3 menyederhanakan permasalahan dengan mengganti himpunan *fuzzy* dengan *fuzzy* tunggal. Hal ini karena pada penerapan yang sebenarnya, masukan akan bersifat *crisp* yang diperlukan sebagai *fuzzy* tunggal. Penafsiran grafis Lemma 3 pada Gambar 2.11 menjelaskan hal ini. Dari aturan kontrol R_1 dan R_2 , dan dengan menerapkan fungsi implikasi operasi mini Mamdani (R_C), produk kartesian operasi minimum dan penggunaan Lemma 3 diperoleh faktor pembobot :

$$\alpha_1 = \mu_{A_1}(X_o) \wedge \mu_{B_1}(Y_o) \quad (2-28)$$

$$\alpha_2 = \mu_{A_2}(X_o) \wedge \mu_{B_2}(Y_o) \quad (2-29)$$



Fungsi keanggotaan sinyal atur adalah fungsi keanggotaan yang didapat dari persamaan di atas merupakan fungsi keanggotaan untuk sinyal atur tersebut. Dengan fungsi keanggotaan itu akan didapat nilai dari sinyal atur yang didapat dari proses fuzzifikasi. (Harris, 1993)



Gambar 2.11 Penafsiran grafis Lemma 3
(Chien Lee, 1990)

2.1.4.6. Defuzzifikasi

Defuzzifikasi merupakan kebalikan dari fuzzifikasi, yaitu mentransformasikan data *fuzzy* ke bukan *fuzzy*, atau pemetaan dari ruang aksi atur *fuzzy* ke aksi atur *crisp*. Defuzzifikasi dinyatakan sebagai :

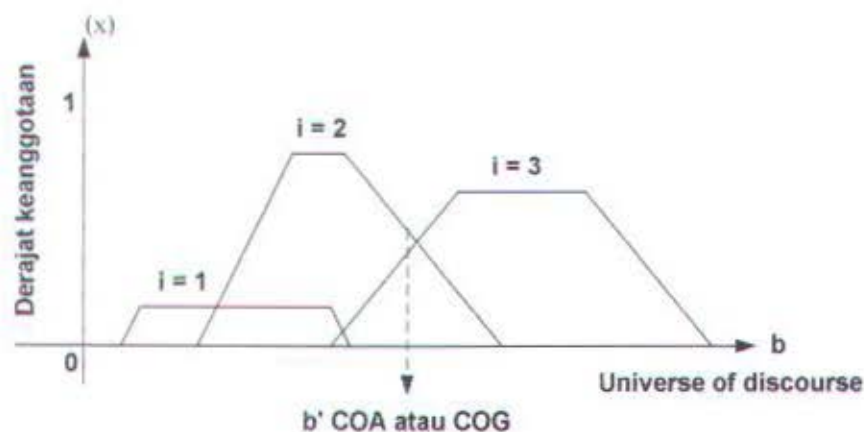
$$Z_o = \text{mem-fuzzikan } (Z)$$

Secara garis besar strategi defuzzifikasi bertujuan menghasilkan aksi atur nyata yang dapat merepresentasikan distribusi dari aksi atur masing-masing kaidah atur, dalam hal ini ada beberapa metoda yang dapat digunakan.



Salah satu metoda yang sering digunakan adalah metoda yang dikenal dengan metoda **COA** (*Centre of Area*) atau metoda titik pusat. Secara umum metoda ini didasarkan pada posisi perhitungan dari suatu pembagian sumbu aksis antara separuh area kiri dan separuh area kanan yang ditentukan oleh hubungan fungsi keanggotaan dari aksi atur *fuzzy*. Pada beberapa kasus metoda ini sama dengan metoda *Centre of Gravity* (**COG**). Dengan kata lain metoda ini menghitung titik pusat dari distribusi sinyal atur dari masing-masing kaidah atur. Sebuah contoh sederhana ditunjukkan pada gambar 2.12 yang mengilustrasikan nilai defuzzifikasi COA atau COG dari sebuah single input – single output pada sistem fuzzy yang dibuat dengan 3 aturan inferensi. Secara umum z_o dihitung dengan menggunakan persamaan titik pusat. (*Negnevitsky*)

$$z_o = \frac{\sum_{i=1}^n \mu_z(z_i) \cdot z_i}{\sum_{i=1}^n \mu_z(z_i)} \quad (2-30)$$



Gambar 2.12 Hasil defuzzifikasi dengan menggunakan metoda COA atau COG
(*Patyra, 1996*)



Pada semesta pembicaraan dengan penyokong berhingga (diskrit) dengan jumlah penyokong (z) n persamaan ini dapat digunakan secara langsung karena setiap penyokong, nilai keanggotaannya ditentukan secara eksplisit (pendefinisian numerik). Pada semesta tak hingga (kontinyu) yang didefinisikan secara fungsional z_i sebagai nilai rata-rata fungsi keanggotaan (μ untuk eksponensial, a untuk segitiga dan trapesium) dan $\mu_x(z_i)$ ditafsirkan sebagai nilai keanggotaan z_i setelah dikalikan bobot kontribusi kaidah atur a_i .

2.2. Mekanisme Start Motor Diesel

Motor diesel dapat distart jika poros engkolnya telah diputar oleh sebuah bantuan alat dari luar sehingga untuk menghasilkan langkah daya maka udara yang dikompresikan pada TMA sampai pada suatu tekanan tertentu, yang apabila bahan bakar diinjeksikan akan menghasilkan proses penyalaan. Ada dua persyaratan penting yang harus dipenuhi untuk start yang pasti dan cepat, yakni : kecepatan motor induk yang cukup dan perbandingan kompresi yang tepat.

Jikalau motor induk diputar sangat lambat, kebocoran kecil yang tidak dapat dihindarkan akan melintasi cincin torak dan mungkin melalui katup masuk dan buang yang akan memungkinkan sebagian dari udara lari dari silinder selama langkah kompresi. Sehingga hal ini dapat menurunkan tekanan kompresi dan suhu pada akhir langkah di bawah yang diperlukan untuk menyalakan bahan bakar yang diinjeksikan. Oleh sebab itu, terdapat kecepatan minimal yang harus dicapai oleh motor induk sebelum terjadi penyalaan dan motor dapat memulai



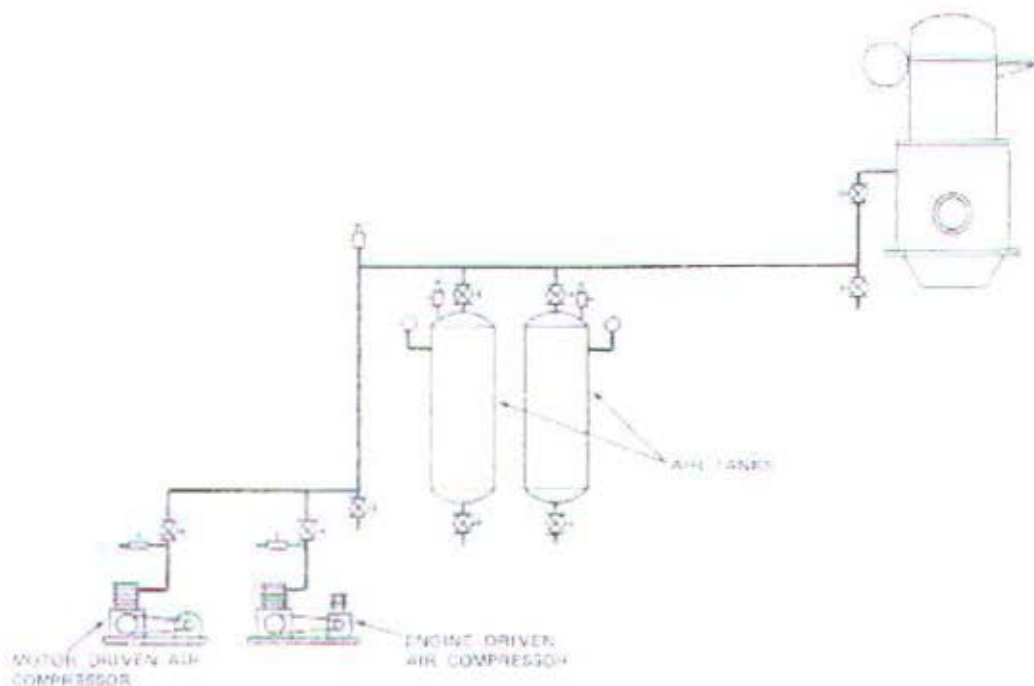
pembakaran. Kecepatan menstart tergantung pada jenis dan ukuran motor, keadaannya dan suhu udara sekeliling. Sedangkan jikalau perbandingan kompresi tidak cukup tinggi, maka suhu akhir dari pengisian udara tekan juga akan terlalu rendah untuk penyalaan. (*Maleev, 1986*)

Seperti yang diketahui bahwa proses start dari motor diesel memiliki beberapa macam metoda pengoperasiannya. Entah itu distart dengan menggunakan tangan, dengan motor listrik, dengan mesin bensin bantu, dengan udara tekan, atau dengan peluru ledak (*explosive cartridge*). Yang kesemuanya itu tergantung dari jenis dimensi dari motor itu sendiri dan juga tergantung dari perancangannya. Umumnya di kapal, penggerak utamanya yang berupa motor diesel menggunakan udara tekan untuk kepentingan menstart. Beberapa alasannya adalah bahwa udara tekan murah untuk diproduksi, mudah disimpan dan sebagai gas, berkelakuan selama ekspansi mirip dengan gas pembakaran dalam silinder.

Metoda start dengan udara tekan terutama pada motor diesel besar memerlukan penggunaan energi yang cukup besar dalam waktu yang singkat. Penekanan udara ke dalam tangki dan penggunaan udara dari tangki dapat memberikan energi yang diperlukan sejumlah berapa pun yang dikehendaki. Tangki udara tekan dan penggerak katup start seluruhnya tidak terlalu menambah besar dan berat kepada motor, dan metoda ini umumnya digunakan pada motor diesel besar.



Tekanan udara biasanya sekitar 150 sampai 300 psi. Motor injeksi udara mempunyai kompresor udara tekanan tinggi, dan untuk memperkecil ukuran tangki udara, digunakan tekanan udara sampai 500 atau bahkan sampai 700 psi. Volume tangki udara yang diperlukan untuk menstart motor dapat diambil sebesar lima belas sampai dua puluh lima kali lipat perpindahan torak total untuk motor kecil, sedangkan pada motor-motor besar diperlukan sekitar tujuh sampai sepuluh kali lipat perpindahan torak. Dalam gambar di bawah ini ditunjukkan sebuah bentuk sistem udara start dalam bentuk skema. Di dalam beberapa instalasi dua atau lebih tangki udara dapat digunakan, jumlahnya tergantung atas ukuran dari *engine* dan lamanya waktu yang diperlukan untuk proses starting.



Gambar 2.13. Sebuah bentuk sistem udara start (*Toboldt, 1983*)



Udara tekan yang digunakan untuk start dapat dikembalikan dalam jangka waktu yang relatif lama, setelah motor distart. Oleh sebab itu, kompresor udara bisa kecil dan tidak memerlukan banyak daya. Kompresor dapat digerakkan langsung dari motor atau dari sumber daya terpisah, misalnya motor bakar kecil yang distart dengan tangan atau motor listrik.

Untuk katup pemasukan udara start adalah dari jenis popet dan dipasangkan dalam kepala silinder. Dikarenakan ada beberapa metoda pengoperasian katup ini, maka starter udara dapat dibagi menjadi tiga kelompok umum, yaitu : (1) dioperasikan dengan tangan, untuk digunakan pada motor silinder tunggal, (2) dioperasikan dengan nok, untuk motor dengan ukuran sedang, dan (3) dioperasikan dengan udara tekan oleh katup pandu yang dioperasikan dengan nok, untuk semua motor besar.

2.2.1. Kedudukan Menstart

Sebuah motor empat langkah dapat distart dari sembarang kedudukan dari poros engkol hanya kalau mempunyai lima silinder atau lebih, yang masing-masing dilengkapi dengan katup penstarter udara. Semua motor empat langkah lainnya harus diputar lebih dahulu ke kedudukan sedemikian rupa sehingga salah satu dari torak berada pada langkah menurun, sedikit melampaui titik mati dan siap untuk mengambil udara penstarter. Sebaliknya, motor dua langkah dapat distart dari sembarang kedudukan poros engkol, kalau mempunyai tiga silinder atau lebih, semuanya dengan katup penstarter udara.



Dalam motor kecil, poros engkol diputar ke kedudukan yang diperlukan dengan memutar roda gila, baik dengan tangan langsung atau dengan batang yang disisipkan ke dalam lubang pada pelek roda gila. Motor besar diputar dengan roda cacing yang dihubungkan selama operasi ini dengan potongan roda gigi pada pelek roda gila atau dengan mekanisme lidah, atau dongkrak. Dalam motor besar, roda cacing dapat dioperasikan oleh motor listrik atau motor udara tekan dan mekanisme lidah oleh motor udara tekan ulak-alik. (*Maleev, 1986*)

2.2.2. Pemeriksaan Sebelum Start Motor Diesel

Seperti yang telah disinggung pada Bab Pendahuluan, yakni sebelum motor diesel dilakukan proses start ada beberapa tahapan-tahapan yang harus dilakukan terlebih dahulu oleh si operator. Secara garis besar, beberapa hal yang harus dilakukan umumnya yaitu (*Maleev, 1986 ; Arismunandar, 1993*):

- 1) Dilakukan pemeriksaan terhadap sistem pelumasan, di mana hal tersebut meliputi : pemeriksaan terhadap pompa, pipa, katup dan terutama sekali jumlah minyak pelumas. Pemeriksaan jumlah minyak pada motor dapat dilakukan dengan batang pengukur minyak pelumas. Tariklah batang tersebut dari dalam motor dan bersihkan dengan lap yang bersih, kemudian kembalikan ke tempat semula. Tariklah batang tersebut sekali lagi, dan periksalah apakah batang tersebut yang dibasahi oleh minyak pelumas sampai batas yang diminta. Apabila tidak, tambahkan minyak pelumas, tetapi jangan melampaui batas.



Pastikan bahwa minyak terpenuhi dengan cukup pada setiap tempat yang memerlukan dan juga pastikan persediaan minyak bersih tersedia dengan cukup. Tidak luput juga dilakukan pemeriksaan terhadap tekanan dan temperatur minyak pelumas.

- 2) Dilakukan pemeriksaan terhadap sistem pendinginan, di mana hal tersebut meliputi : pemeriksaan terhadap pompa, katup dan pipa saluran hisap harus dibuka untuk memberikan air di dalam motor sebelum menstart, pemeriksaan juga dilakukan pada jaket (apakah terjadi kebocoran), pemeriksaan juga untuk jumlah dari sirkulasi air pendingin, apakah jumlahnya sesuai dengan yang tercantum di dalam buku pedoman. Jangan sampai kekurangan air pendingin. Dan dilakukan juga pemeriksaan terhadap temperatur air pendingin baik pada air laut maupun pada air tawar (jika memakai pendingin air tawar).
- 3) Dilakukan pemeriksaan terhadap sistem bahan bakar, di mana hal tersebut meliputi : pemeriksaan terhadap pompa, katup, pipa dan penyediaan bahan bakar di dalam tangki harus cukup tersedia. Jumlah bahan bakar harus dapat mencukupi kebutuhan sehingga motor tidak akan mati karena kehabisan bahan bakar. Pemeriksaan juga dilakukan pada katup pengaman yang biasanya dipasang pada tiap kepala silinder. Katup ini disetel untuk meletup pada kira-kira 750 sampai 1250 psi, tergantung pada tekanan maksimum yang dibolehkan dalam motor. Katup dihadapkan pada gas bertemperatur tinggi dan



mempunyai kecenderungan untuk macet. Pemeriksaan dapat dilakukan dengan menekan pegas menggunakan batang pengungkit atau dengan melepas sekrup tutup dan mengeluarkan katup untuk diperiksa.

- 4) Dilakukan pemeriksaan terhadap sistem start, di mana hal tersebut meliputi : pemeriksaan terhadap kompresor, katup, pipa, saringan, mur, baut, sambungan-sambungan, juga pemeriksaan terhadap udara dalam tangki, apakah tekanannya cukup. Kalau tidak, maka dilakukan pengisian. Sistem penstarter udara dari tangki sampai ke katup pengendali utama penstarter harus dibuka, setelah diperiksa bahwa katup pengendali utama tertutup.
- 5) Dilakukan pemeriksaan terhadap kondisi beban dari motor. Beban motor harus diputuskan atau motor tidak boleh terbebani dalam keadaan start. Yaitu dengan cara : saklar harus dibuka kalau motor menggerakkan generator, atau kopling harus berada dalam kedudukan netral kalau penggeraknya melalui kopling gesek.

2.2.3. Menstart Motor Diesel

Setelah beberapa hal persiapan sebelum start sudah dilakukan dengan baik dan benar, maka penstarteran dengan udara tekan akan cukup mudah dilaksanakan. Namun, ada tiga hal yang harus diperhatikan dalam menstart, yakni : *Pertama*, katup penstarter udara utama dibuka dan batang penstarter diatur menurut petunjuk yang diberikan dalam buku pedoman. *Kedua*, motor diawasi ; tidak boleh digunakan udara yang tidak diperlukan. Pada tanda pertama dari pembakaran udara harus dimatikan dan katup ventilasi dibuka. Sebuah motor



dalam keadaan yang baik biasanya mulai penyalaan di antara putaran kedua dan keempat dari poros engkol. *Yang Ketiga*, kalau motor gagal untuk start setelah empat atau lima putaran, berarti ada sesuatu yang salah. Pemutaran tidak berguna dari motor harus dihentikan dan penyebab gangguan diselidiki. (*Maleev, 1986*)

2.2.4. Pemanasan Setelah Start Motor Diesel

Setelah motor distart, sebelum dibebani, harus dibiarkan tanpa kerja untuk beberapa menit (biasanya sekitar 5 menit) sampai setiap bagian motor dan air atau minyak pelumas mencapai temperatur kerjanya yang normal. Beberapa hal yang dilakukan adalah (*Maleev, 1986 ; Arismunandar, 1993*) :

- 1) Tekanan minyak pelumas. Setiap motor diesel diberitahukan berapa besar tekanan minyak pelumasnya yang normal, umumnya berkisar di antara 2 sampai 4 kg/cm². Segera setelah motor dapat distart, tekanan minyak pelumasnya dapat mencapai bilangan yang tinggi, karena kekentalannya masih tinggi. Tetapi akan berkurang apabila motor sudah panas. Kalau tekanannya tidak naik, segera matikan motor dan diselidiki sebabnya. Periksa juga apakah ada silinder yang terlalu cepat panas, yang menunjukkan adanya torak yang tidak terlumasi, dan dengarkan kalau ada bantalan pena torak atau pena engkol yang tidak terlumasi. Kalau ada bagian bergerak yang tidak cukup mendapatkan minyak pelumas, dapat mengakibatkan kerusakan kritis.
- 2) Amati sistem air pendingin keseluruhan untuk mengetahui apakah pompa bekerja dan terdapat air yang cukup. Lihat juga apakah



temperatur air menanjak dengan baik, dan atur aliran air untuk menyesuaikan.

- 3) Amati apakah terdengar bunyi dan getaran. Biasanya motor berbunyi keras pada pemulaan start, tetapi bunyi tersebut berangsur-angsur akan menjadi lunak setelah motor menjadi panas.
- 4) Amati juga pada warna dan suara gas buang yang terjadi untuk mengetahui keadaan yang baik. Pengamatan ini harus diulangi setelah beban disambungkan. Warna dari gas buang yang terjadi dapat menjelaskan banyak hal.
- 5) Periksa apakah terjadi kebocoran air atau minyak pelumas.

Tindakan pengamatan ini selama lima menit pertama setelah menstart harus menjadi kebiasaan bagi operator di kamar mesin. Jika operator di kapal mengikuti prosedur yang ada maka dipastikan tidak akan terjadi permasalahan dan juga dapat mencegah operasi yang tidak benar. Hal ini didasarkan pada kenyataan bahwa motor diesel memerlukan perhatian yang layak pada saat yang tepat. Juga didasarkan pada kenyataan yang telah diketahui bahwa motor diesel harus dioperasikan dengan baik dalam lima menit atau terdapat satu kelainan yang harus ditemukan dalam lima menit tersebut.

Tetapi, perlu dicatat bahwa pengamatan tertentu harus dilakukan meskipun setelah perioda pemanasan lima menit. Yaitu, kalau terdapat kebocoran pada jaket air, katup injeksi, katup udara dan sebagainya, mereka mungkin tidak terlihat



sampai pemuaian sepenuhnya dari bagian yang bersangkutan terjadi setelah motor beroperasi untuk waktu yang lebih lama dalam beban normal. Tidak boleh ada kebocoran jenis apapun juga, kalau mereka tidak dapat dihentikan sementara motor berjalan, motor harus dihentikan dan tidak boleh distart kembali sampai kerusakan diperbaiki.



B A B III

PEMODELAN SISTEM DAN IMPLEMENTASI LOGIKA FUZZY

3.1. Tinjauan Umum

Seperti yang telah disinggung pada Bab II bahwa sebelum dilakukan proses *starting* pada motor diesel di kapal ada beberapa tahapan yang harus dilakukan oleh si operator kapal di kamar mesin. Secara garis besarnya tahapan-tahapan tersebut sangat berhubungan dengan sistem-sistem pendukung motor induk. Dari beberapa sistem pendukung motor induk yang akan dilakukan pemeriksaan tersebut, yang paling dominan dan menjadi perhatian oleh kebanyakan si operator di kapal adalah pada sistem pelumasan dan pada sistem pendinginan. Sebagai studi kasus dalam pembahasan Tugas Akhir ini, diambil data-data mengenai permasalahan di atas pada kapal ferry penyeberangan milik P.T. Adhiguna Citra Utama dengan nama kapal K.M.P. Adhiswadharma III yang beroperasi di dermaga Ujung – Kamal.

Kedua sistem itulah yang paling diperhatikan di kapal sebelum dilakukannya proses *starting*. Hal itu dikarenakan sebelum motor induk distart, kedua sistem tersebut biasanya harus sudah menunjukkan dalam keadaan baik. Untuk mengetahuinya maka harus dijalankan terlebih dahulu yaitu dengan menyirkulasikan media fluida dari masing-masing sistem tersebut. Dengan tujuan



untuk mengetahui apakah semua komponennya dan aliran fluida tersebut memang benar-benar sudah siap digunakan atau terjadi masalah pada kedua sistem tersebut.

Penekanan yang menjadi perhatian terhadap kedua sistem pendukung motor induk tersebut di atas bukan berarti mengacuhkan beberapa sistem pendukung motor induk yang lain, misalkan : sistem bahan bakar, sistem start dan lain-lainnya. Namun, dalam hal ini kebanyakan fakta di kapal telah menunjukkan bahwa kedua sistem pendukung motor induk, yakni : sistem pelumas dan sistem pendingin, adalah menjadi perhatian sebagai yang dominan. Di samping itu juga di dalam pembahasan Tugas Akhir ini, permasalahan yang dibuat hanya dibatasi pada dua sistem tersebut (sistem pelumas dan sistem pendingin). Jadi, semua yang berhubungan dengan mekanisme pengaturannya akan di titikberatkan pada dua sistem tersebut.

3.2. Model Tahapan-tahapan Persiapan Sebelum Start

Sebelum dibuat suatu model dari tahapan-tahapan persiapan sebelum start terhadap motor induk di kapal penyeberangan maka dilihat terlebih dahulu beberapa parameter-parameter yang digunakan dalam pembahasan di sini. Parameter-parameter tersebut adalah :

- 1) Indikator tekanan pada *pressure gage*

Indikator ini nantinya akan ditunjukkan dalam sebuah angka pada manometer. Indikator tersebut diamati setelah pompa dijalankan, baik



pada pelumasan, pendinginan maupun juga pada tangki botol angin. Satuan yang ditunjukkan adalah Kg/cm^2 .

2) Indikator Tekanan pada *compound gage*

Indikator ini juga ditunjukkan berupa angka pada manometer. Biasanya terdapat pada pompa minyak dan air. Hanya saja perbedaan dengan *pressure gage* adalah pada *compound gage* digunakan untuk menunjukkan tekanan hisap pompa sedangkan pada *pressure gage* digunakan untuk menunjukkan tekanan keluar pompa. Angka pada *compound gage* tersebut terdapat dua petunjuk, yakni : petunjuk pertama berupa angka yang menunjukkan tekanan isap dalam satuan cmHg , petunjuk kedua berupa angka yang menunjukkan tekanan isap pompa terjadi kemasukan udara yang ditunjukkan dalam satuan Kg/cm^2 .

3) Indikator Volume

Indikator ini juga penting untuk diperhatikan. Pengamatannya akan ditunjukkan dalam angka pada sebuah batang pengukur pada tiap-tiap tangki, baik itu pada pelumasan, pendinginan maupun pada bahan bakar. Sehingga pengamatan yang dilakukan berdasarkan level tangki tersebut. Satuan yang ditunjukkan berupa liter.

4) Kebocoran

Parameter ini juga penting untuk diketahui dari semua proses yang berjalan. Dikarenakan parameter tersebut berhubungan langsung dengan



beberapa instalasi peralatan di dalam kamar mesin. Sehingga jika terjadi hal tersebut nantinya akan ada suatu tindak lanjut yang harus dilakukan oleh si operator kapal, yakni dicarinya penyebab yang membuat hal itu terjadi. Bisa juga indikasi ini dapat ditunjukkan dengan adanya perubahan tekanan pada peralatan yang bersangkutan.

5) Bunyi dan Getaran

Parameter ini juga berhubungan dengan beberapa instalasi peralatan di kamar mesin sehingga tindak lanjutnya juga sama dengan parameter kebocoran yakni mencari sebab terjadinya. Biasanya untuk mengetahui indikasi ini ditunjukkan berupa perubahan ampere dari peralatan itu sendiri.

6) Penyumbatan

Parameter ini biasanya berhubungan dengan sistem instalasi perpipaan dari tiap-tiap sistem pendukung motor induk. Namun, juga bisa terdapat pada beberapa peralatan dari sistem pendukung tersebut, yakni *cooler* dan *filter*. Adanya indikasi ini dapat ditunjukkan dengan adanya perubahan tekanan.

7) Ampere

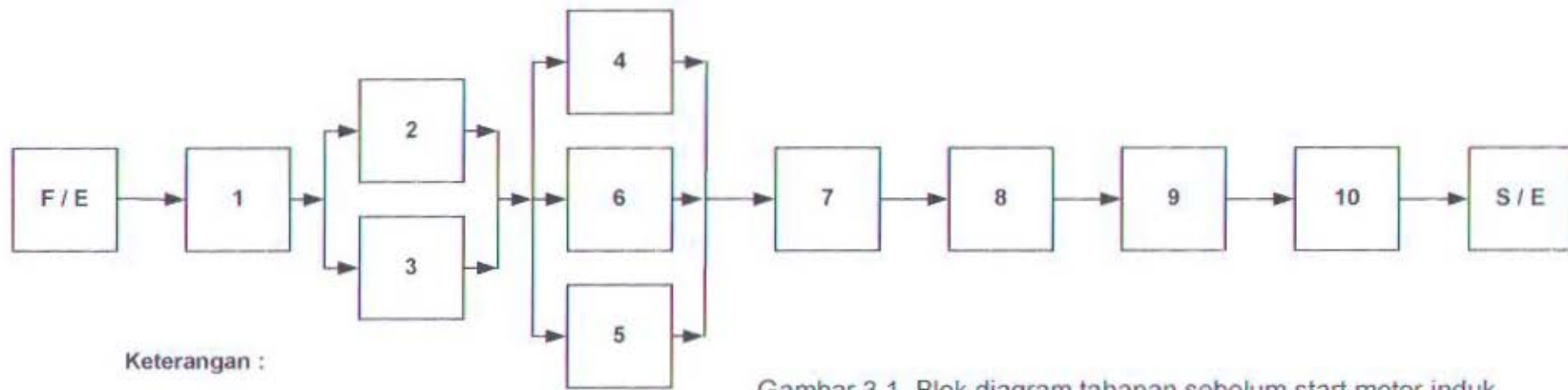
Indikator ini dapat diketahui jika pada peralatan-peralatan seperti pompa (khususnya pada pompa pendingin dan pelumas) terjadi gangguan. Gangguan tersebut bisa terjadi akibat karena adanya bagian-bagian dari



pompa tersebut mengalami masalah, misal : bearing aus, impeler lepas, tidak balans, poros misalignment dan lain-lainnya sehingga biasanya ditandai juga dengan adanya bunyi dan getaran.

Pada gambar 3.1 akan ditunjukkan pemodelan beberapa tahapan yang dilakukan oleh si operator pada kapal ferry penyeberangan. Dalam gambar tersebut telah dilakukan penyusunan sebagai pengaturan terhadap hal-hal yang dilakukan oleh si operator kapal sebelum motor induk distart. Pada beberapa blok dari gambar tersebut nantinya juga ada blok-blok diagram lain yang berada di dalamnya, yaitu blok pemeriksaan terhadap sistem pendukung motor induk. Blok-blok yang berada di dalam tersebut menunjukkan proses tahapan-tahapan sendiri dari beberapa sistem pendukung motor induk yang akan dilakukan oleh si operator kapal. Blok-blok diagram dari tahapan-tahapan setiap sistem pendukung motor induk ditunjukkan pada gambar 3.2, 3.3, 3.4, 3.5 dan 3.6.

Seperti yang telah dijelaskan sebelumnya bahwa dalam pembahasan masalah Tugas Akhir ini, batasan masalah yang diambil hanya melingkupi pada dua sistem pendukung motor induk saja yakni : sistem pendingin dan sistem pelumasan. Dari dua sistem inilah nantinya akan diorientasikan untuk dilakukan pengaturan dari beberapa indikasi yang bisa untuk diatur dengan logika *fuzzy*. Sedangkan untuk beberapa indikasi yang tidak bisa diatur dengan logika *fuzzy* dapat ditunjukkan dengan menampilkan informasi-informasi kepada si operator lewat *windows interface*.

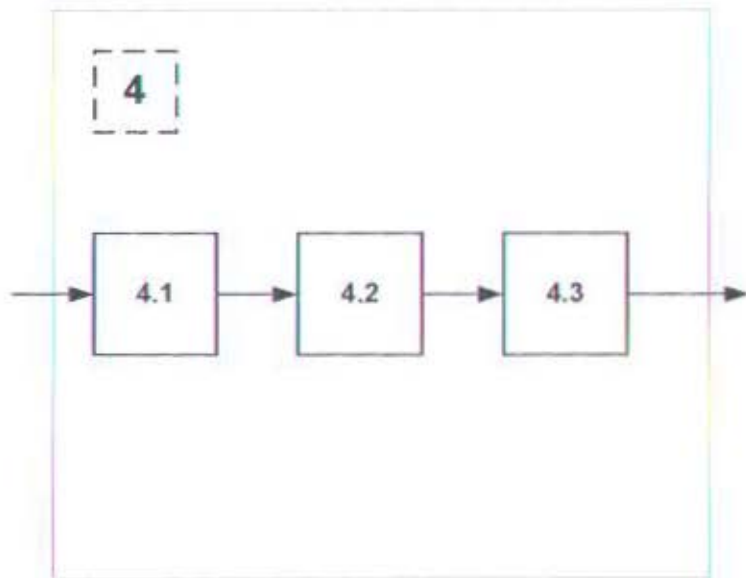


Gambar 3.1 Blok diagram tahapan sebelum start motor induk.

F / E = Motor induk dalam keadaan berhenti atau mati.

1. Hidupkan blower untuk ventilasi di kamar mesin (blower harus selalu dihidupkan meskipun M/E off)
2. Pemeriksaan terhadap bagian-bagian dari M/E yang bergerak (katup, nok, penggerak katup, pompa bahan bakar, sistem injeksi bahan bakar, governor) untuk penyetelan dan pelumasan yang baik.
3. Pemeriksaan terhadap bagian-bagian dari M/E dan permesinan yang tidak bergerak, meliputi mur, baut, sambungan, jaket.
4. Pemeriksaan terhadap sistem pelumasan, meliputi : jumlah oli pada tangki, pompa, tekanan, pipa dan katup.
5. Pemeriksaan terhadap sistem air pendingin, meliputi : sistem pendingin air tawar dan air laut.
 Pada sistem pendingin air tawar meliputi : jumlah level air pada tangki ekspansi, tekanan, pompa, pipa dan katup.
 Pada sistem pendingin air laut meliputi : saluran hisap, tekanan, pompa dan katup.
6. Pemeriksaan terhadap sistem bahan bakar, yaitu meliputi pemeriksaan jumlah bahan bakar pada tangki, pompa, pipa dan katup.
7. Pemeriksaan terhadap sistem udara start, meliputi : tekanan dalam tangki reservoir, pipa, katup start.
8. Pemeriksaan terhadap saluran udara bilas dan buang, meliputi air intake screen dan exhaust shut off valve.
9. Pemeriksaan terhadap beban M/E (harus diputuskan dan M/E tidak boleh dalam keadaan terbebani).
10. Buka katup penstarter udara.

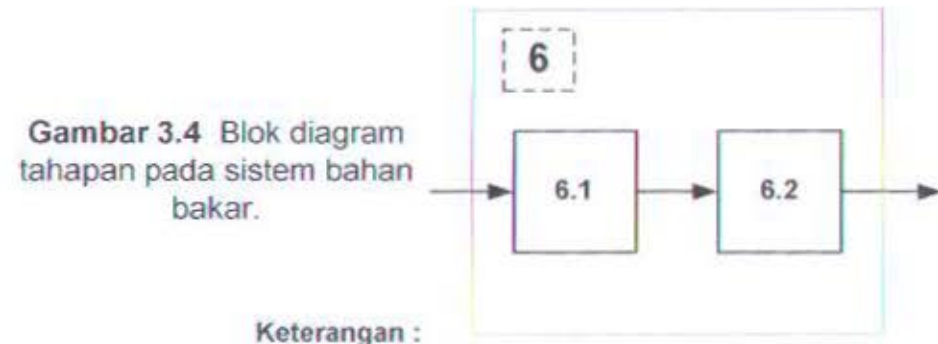
S / E = Proses starting motor induk



Gambar 3.2 Blok diagram tahapan pada sistem pelumas.

Keterangan :

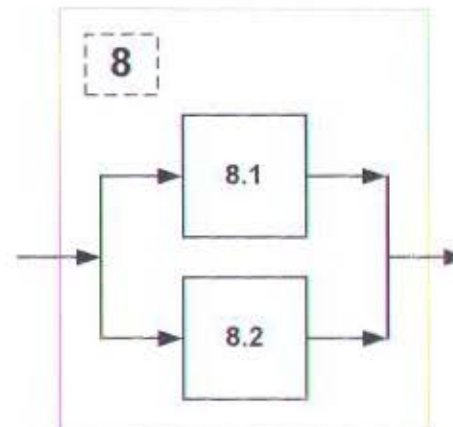
- 4.1. Check volume oli pada tangki.
- 4.2. Jalankan motor pompa stand by.
- 4.3. LO system.



Gambar 3.4 Blok diagram tahapan pada sistem bahan bakar.

Keterangan :

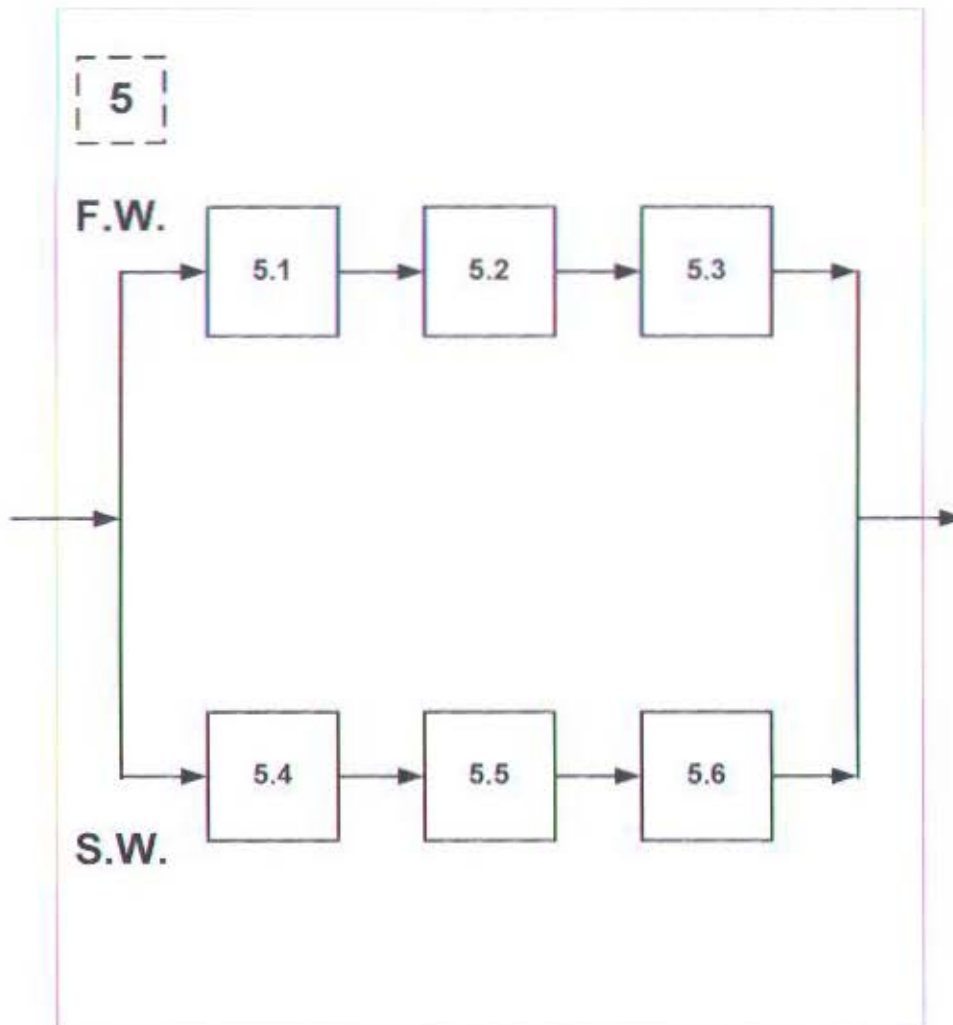
- 6.1. Check penyediaan bahan bakar di dalam tangki.
- 6.2. Kendali pompa bahan bakar pada kedudukan Fuel On.



Gambar 3.6 Blok diagram tahapan pada sistem udara bilas dan buang.

Keterangan :

- 8.1. Check air intake screen.
- 8.2. Buka katup exhaust shut off.

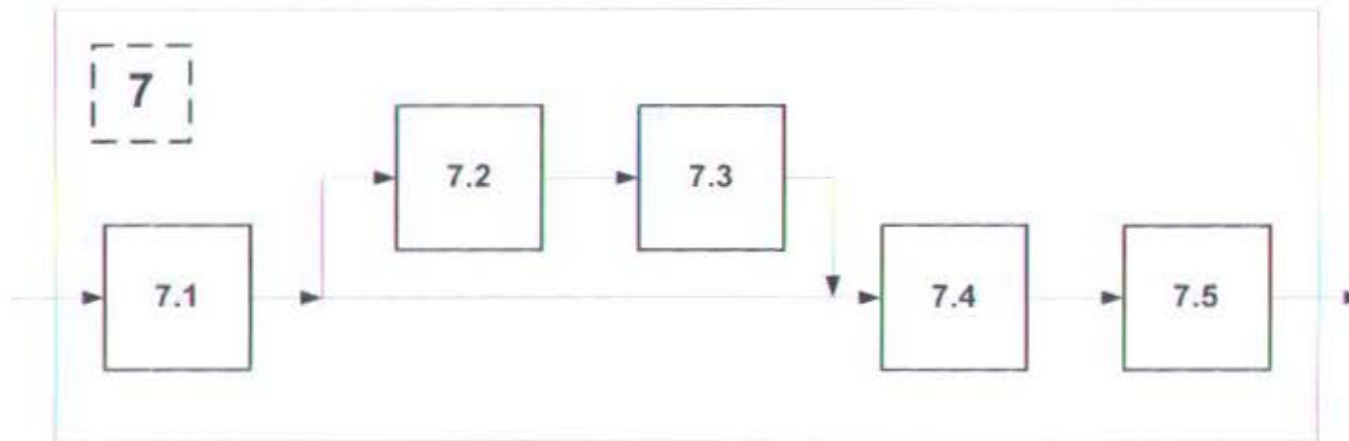


Gambar 3.3 Blok diagram tahapan pada sistem pendinginan.

Keterangan :

- 5.1. Check level volume air tawar pada tangki ekspansi.
- 5.2. Jalankan motor pompa Fresh Water.
- 5.3. FW cooling system.

- 5.4. Buka saluran hisap Sea Water (katup air laut).
- 5.5. Jalankan motor pompa Sea water.
- 5.6. SW cooling system.



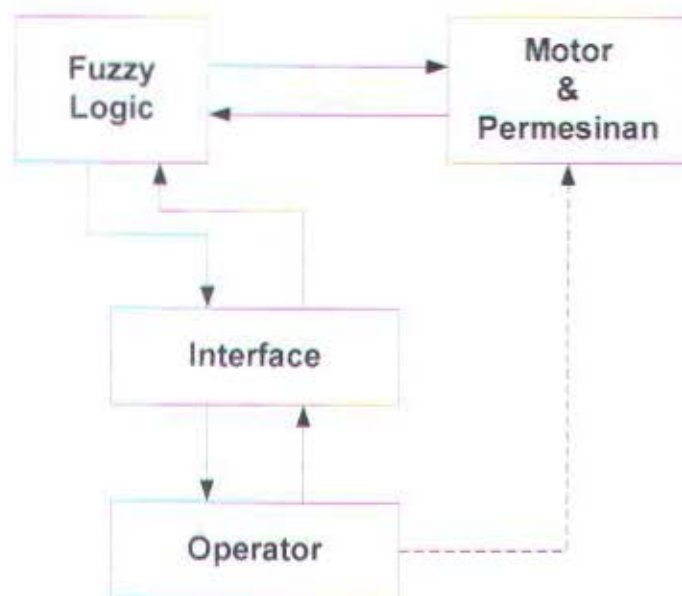
Gambar 3.5 Blok diagram tahapan pada sistem udara start.

Keterangan :

- 7.1. Check tekanan pada botol angin / reservoir (maksimum 29 kg/cm².)
- 7.2. Lakukan pengisian jika terjadi kekurangan.
- 7.3. Check tekanan kembali.
- 7.4. Check apakah terdapat kebocoran / penyumbatan (pada pipa).
- 7.5. Check katup start.

3.3. Arsitektur Sistem Pengaturan

Pada gambar 3.7 akan ditunjukkan suatu diagram arsitektur dari sistem pengaturan yang akan digunakan, yakni :



Gambar 3.7 Arsitektur Sistem Pengaturan.

Pada skema diagram di atas, plant dari sistem yang akan diatur yakni tahapan-tahapan saat F/E (Finish Engine) sampai dilakukan S/E (Start Engine). Namun, sesuai dengan batasan masalah yang diambil dalam pembahasan ini maka plant dari sistem yang akan diatur diambil berupa sistem pelumasan dan sistem pendinginan motor induk saja.

Pertama kali pada *plant* akan terdapat inputan berupa indikasi dari beberapa indikator yang berhubungan dengan sistem yang ada. Indikasi tersebut bisa berupa tekanan, volume, ampere, temperatur dan lain-lainnya. Dari beberapa



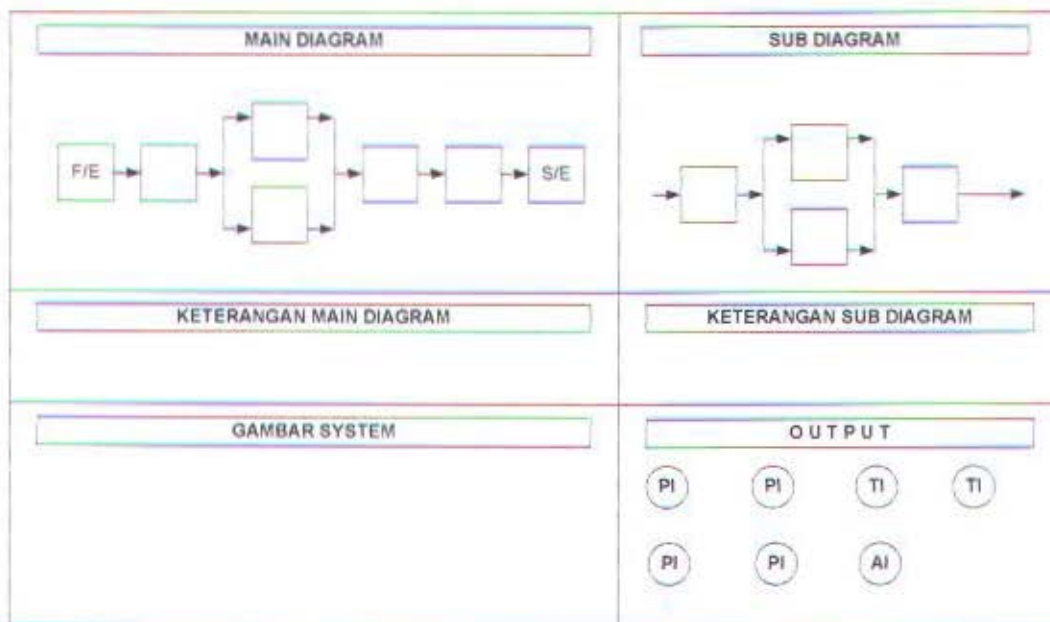
inputan tadi akan dikirim melalui pengontrol logika *fuzzy*. Inputan tersebut akan dikuantisasi. Setelah terkuantisasi maka dilanjutkan pemrosesan data yang masuk dari blok Mesin dan Permesinan. Dalam hal ini data-data pada blok Mesin dan Permesinan dibuat dalam bentuk data base, dikarenakan pembahasan di sini tidak berupa aplikasi langsung di lapangan melainkan berupa studi perencanaan. Di mana inputan tersebut akan diproses berdasarkan aturan-aturan yang telah dibuat dalam sistem pengaturan tersebut. Aturan-aturan yang dibuat tentunya berdasarkan pengalaman dari si operator kapal yang dibuat dalam bentuk data base. Sehingga setelah diketahui hubungan antara input – aturan – output maka nantinya akan dikembalikan lagi pada *plant* yang akan diatur, yakni berupa informasi-informasi yang harus dilakukan oleh si operator kapal nantinya jika *plant* tersebut terjadi gangguan ataupun memberikan informasi bahwa sistem tersebut masih berjalan dengan normal. Kesemua informasi tersebut akan ditampilkan melalui *windows interface* lewat layar monitor sehingga nantinya operator kapal akan menjadi lebih cepat mengetahui dan bertindak jika ada *malfunction* dari *plant* yang diatur. Yaitu bisa langsung menuju ke Mesin dan Permesinan yang mengalami gangguan untuk diperbaiki.

3.4. Model Windows Interface

Setelah didapatkan model-model dari *plant* yang akan diatur dengan logika *fuzzy*, khususnya pada tahapan sistem pelumas dan sistem pendingin saat F/E sampai dilakukan S/E, maka perencanaan untuk ditampilkan pengaturan dari *plant* tersebut pada *windows interface* lewat layar monitor dapat diperlihatkan pada gambar 3.8.



Pada bentuk tampilan windows interface tersebut terdapat beberapa bagian yang menampilkan informasi berlainan dari satu dengan yang lain. Bagian-bagian tersebut terdiri dari : Main diagram, Keterangan main diagram, Sub diagram, Keterangan sub diagram, Gambar system dan Output.



Gambar 3.8 Model plant pada windows interface.

Pada *Main Diagram* akan diperlihatkan skema diagram utama dari *plant*, yakni tahapan-tahapan saat F/E sampai S/E dari motor induk di kapal. Untuk windows *Sub Diagram*, diperlihatkan skema bagian yang terdapat di dalam setiap blok pada *Main Diagram*. Sedangkan windows Keterangan, baik pada *main diagram* dan *sub diagram*, berisi beberapa informasi mengenai penjelasan langkah-langkah pada masing-masing diagram. Untuk windows *Gambar system*, di sini akan ditunjukkan skema aliran dari sistem yang ada. Dikarenakan dalam batasan masalah di sini adalah pada sistem pelumas dan sistem pendingin maka



yang akan muncul pada window ini hanya skema aliran dari kedua sistem tadi. Sedangkan pada windows *Output* akan ditampilkan informasi dari beberapa indikasi yang berkaitan dengan *Input* yang terjadi pada setiap sistem yang akan ditampilkan. *Output* tersebut berupa indikator yang ada pada sistem yang diatur proses tahapannya.

3.5. Perancangan Pengatur Logika Fuzzy

Berdasarkan metodologi perancangan suatu pengaturan dengan logika *fuzzy* yang telah dibahas pada bab 2, maka dilakukan perancangan suatu pengaturan logika *fuzzy* untuk sistem pengaturan tahapan-tahapan dari operasional permesinan kapal ferry penyeberangan (khususnya pada motor induk) saat *Main Engine* dalam keadaan berhenti hingga dilakukan proses *starting*. Pertama-tama yang harus dilakukan adalah menentukan sistem yang akan diatur kemudian tinjauan secara kualitatif terhadap objek yang akan diatur dan pengaturannya untuk menentukan bentuk basis pengetahuan. Setelah itu menentukan semesta pembicaraannya.

Proses untuk menentukan basis pengetahuan dalam pengatur logika *fuzzy* adalah sebagai berikut :

1. Menggambarkan strategi yang dilakukan operator manusia dalam mengoperasikan suatu sistem pengaturan.
2. Menentukan bentuk-bentuk pengertian linguistik dalam sistem tersebut.
3. Menentukan model untuk memprediksi hasil operasi.



4. Mengkonversikan strategi linguistik operator manusia menjadi aturan dasar.

Hasil rancangan diwujudkan dalam suatu algoritma pengaturan yang direalisasikan dalam bentuk perangkat lunak pada komputer. Untuk proses selanjutnya akan dilakukan simulasi pada sistem telah ditentukan.

Sebelum akan dilakukan penyusunan algoritma pengaturan *fuzzy*, sebelumnya aksi pengaturannya yang dinyatakan dalam bahasa natural ditentukan terlebih dahulu. Untuk unit pengatur *fuzzy* yang digunakan adalah metoda statik, yaitu sifat dari fungsi keanggotaannya mempunyai rentang kerja yang tetap.

3.5.1. Penentuan Variabel Masukan dan Keluaran Fuzzy

Dalam merancang pengatur *fuzzy*, yang perlu diperhatikan adalah variabel masukan dan keluaran yang ditentukan. Variabel masukan tersebut berupa petunjuk dari indikasi-indikasi Mesin dan permesinannya yang ada di kamar mesin, di mana dalam hal ini dibuatkan dalam bentuk data base.

- Untuk inputan pada bagian sistem pelumasan adalah :

- Volume oli pada sump tank (maksimum 2000 liter).
- Switch pompa dan Rpm (rpm 1150).
- Tekanan hisap pompa (normal sekitar 20 – 30 cmHG).
- Tekanan hisap kemasukan udara (ditunjukkan berupa indikasi tekanan berupa satuan kg/cm^2 , maksimum indikasinya 4 kg/cm^2).



- Temperatur dari oli (normal sekitar 35 °C).
- Perubahan tekanan keluar setelah melewati filter dan setelah atau tidak melewati cooler.
- Ampere dari pompa (maksimum 11 ampere, normal sekitar 7-8 A).

Sedangkan untuk output yang dipakai adalah :

- Untuk output pada bagian sistem pelumasan adalah :

- Pengisian oli pada sump tank.
- Kondisi dari pompa.
- Tekanan keluaran pompa (normal sekitar 3,5 kg/cm²).
- Kondisi kekentalan oli.
- Kondisi tekanan keluar setelah melewati filter dan setelah atau tidak melewati cooler.
- Bunyi dan getaran yang terjadi pada pompa (universe of discourse ditunjukkan dengan prosentase terhadap putaran pompa, rpm pompa 1150).
- Panas dari pompa (universe of discoursenya ditunjukkan berupa temperatur, normal sekitar 35-36 °C).

3.5.2. Strategi Pengaturan Pengatur Fuzzy

Untuk pemakaian pengatur fuzzy terdapat beberapa parameter yang menentukan kemampuan dari pengatur. Parameter-parameter tersebut adalah :

- a. Pemilihan variabel masukan dan keluaran.
- b. Himpunan fuzzy dan fungsi keanggotaan.

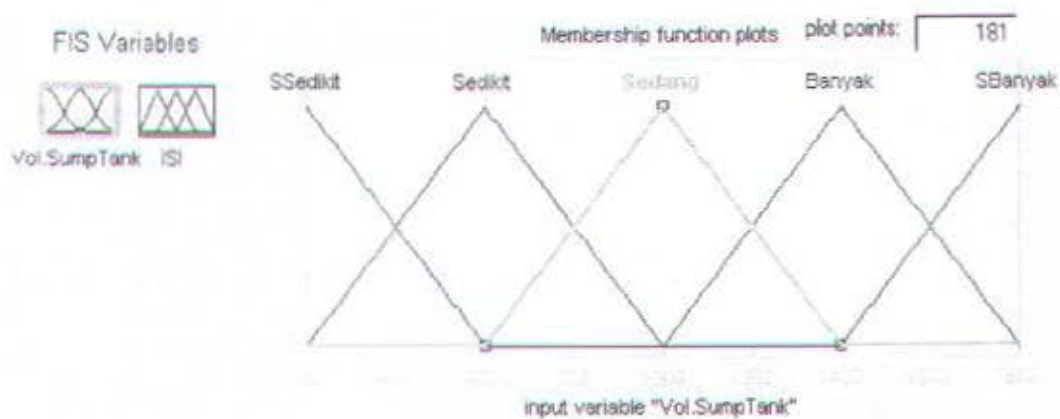
- c. Operator inferensi dan implikasi.
- d. Penyusunan basis data.
- e. Operasi defuzzifikasi.

3.5.3. Basis Data

Penyusunan basis data berarti mendefinisikan himpunan-himpunan fuzzy dalam daerah masukan dan keluaran. Hal ini ditekankan pada pembagian ruang masukan dan keluaran dengan sejumlah nilai variabel linguistik (himpunan fuzzy) yang menggunakan fungsi keanggotaan tertentu.

- Untuk inputan dan outputan pada bagian sistem pelumasan, variabel linguistiknya adalah :
 - Inputan : Volume oli pada sump tank dan ouputan : lakukan pengisian

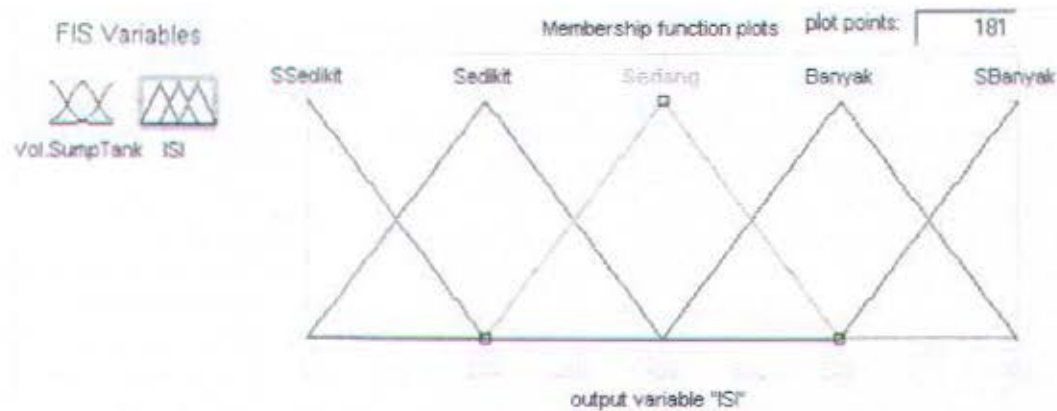
Input : Volume oli (liter)



Gambar 3.9 Fungsi keanggotaan volume oli.



Ouput : Lakukan pengisian (dalam liter)



Gambar 3.10 Fungsi keanggotaan lakukan pengisian pada sump tank.

➤ Inputan : Switch pompa dan rpm Outputan : Kondisi pompa

Input : switch pompa dan rpm

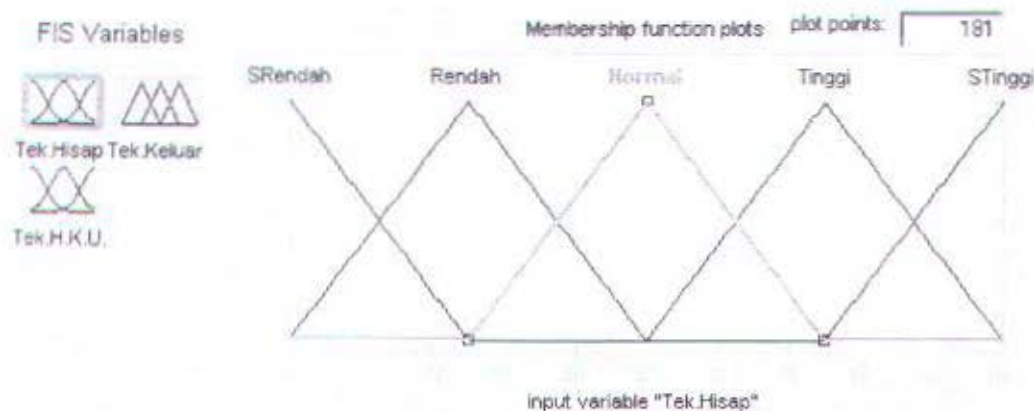
Untuk switch pompa anggotanya adalah : On dan Off

Untuk rpm anggotanya adalah : 0 dan ≤ 1150

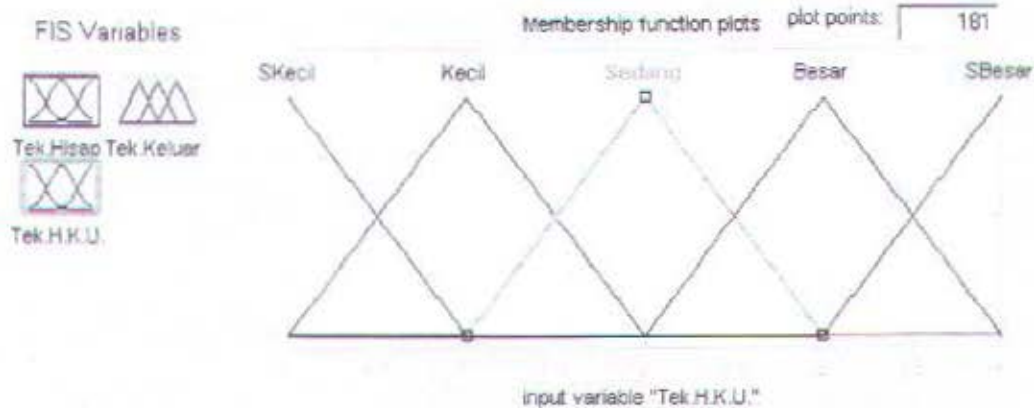
➤ Inputan : Tekanan hisap dan tekanan hisap kemasukan udara (P1) dengan

Outputan : Tekanan keluar (P2)

Input : Tekanan Hisap (cmHg) dan Tekanan Hisap Kemasukan Udara (Kg/cm^2)

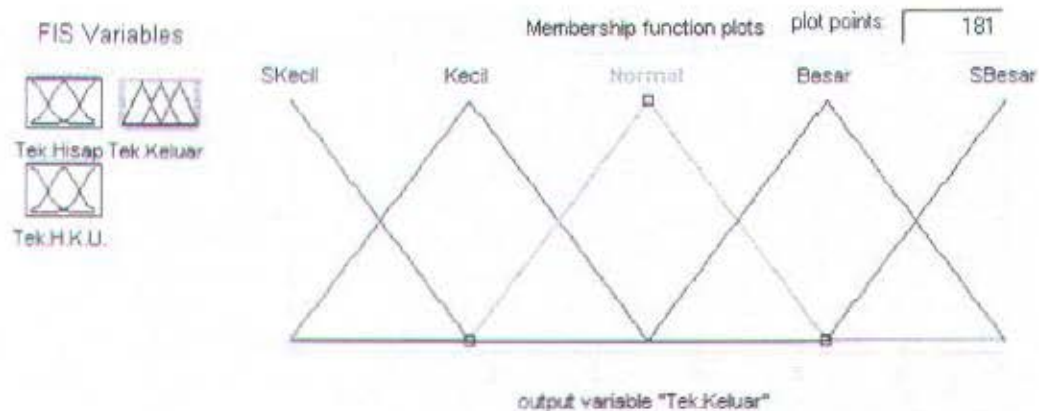


Gambar 3.11 Fungsi keanggotaan tekanan hisap pompa oli.



Gambar 3.12 Fungsi keanggotaan tekanan hisap kemasukan udara pompa oli.

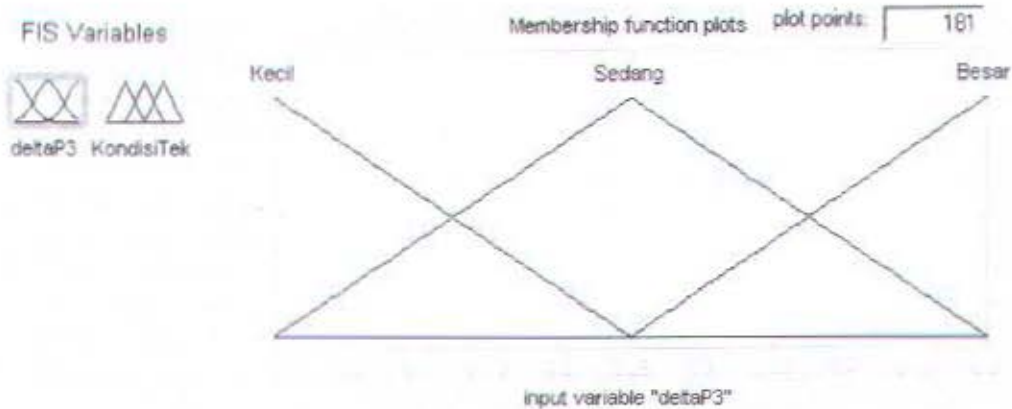
Output : Tekanan Keluar (Kg/cm^2)



Gambar 3.13 Fungsi keanggotaan tekanan keluar pompa oli.

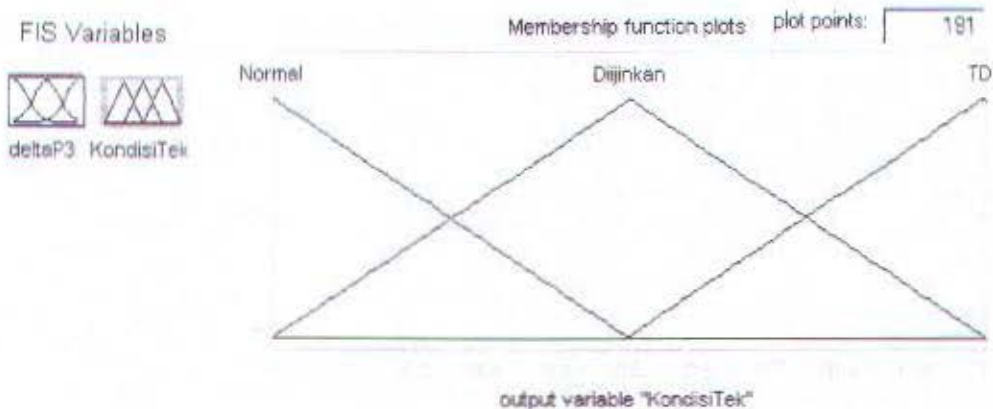
- Inputan : Perubahan tekanan keluar setelah melewati filter (ΔP_3) dengan
- Outputan : Kondisi tekanan keluar setelah melewati filter (P_3)

Input : ΔP_3 (dalam Kg/cm^2)



Gambar 3.14 Fungsi keanggotaan delta P3 oli.

Ouput : Kondisi Tekanan Keluar P3

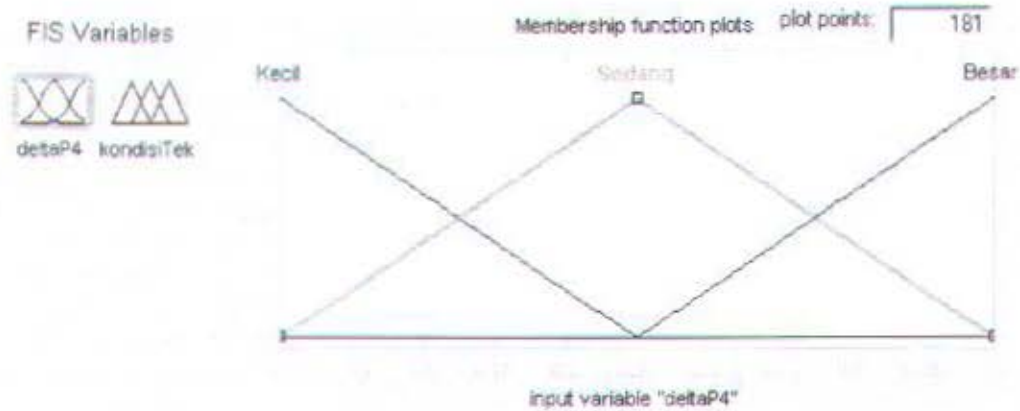


Gambar 3.15 Fungsi keanggotaan kondisi tekanan keluar P3 oli.

- Inputan : Perubahan tekanan keluar setelah atau tidak melewati cooler (ΔP_4) dengan Outputan : Kondisi tekanan keluar setelah atau tidak melewati cooler (P4)

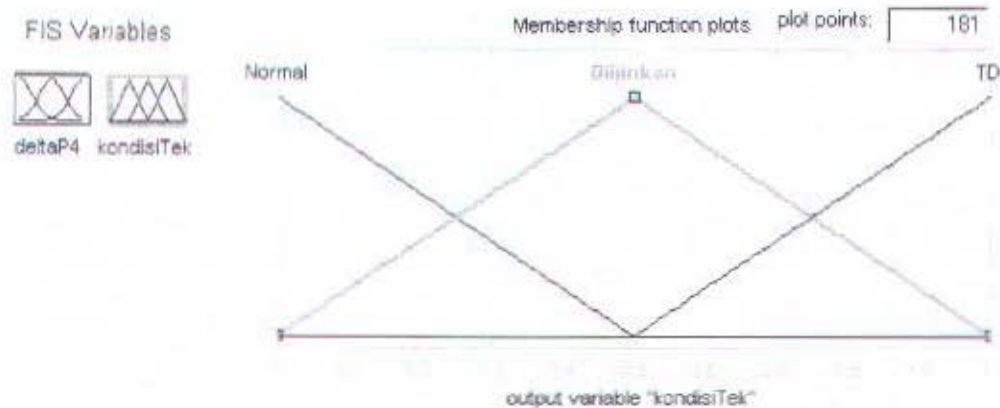


Input : ΔP_4 (dalam Kg/cm^2)



Gambar 3.16 Fungsi keanggotaan delta P4 oli.

Ouput : Kondisi Tekanan Keluar P4

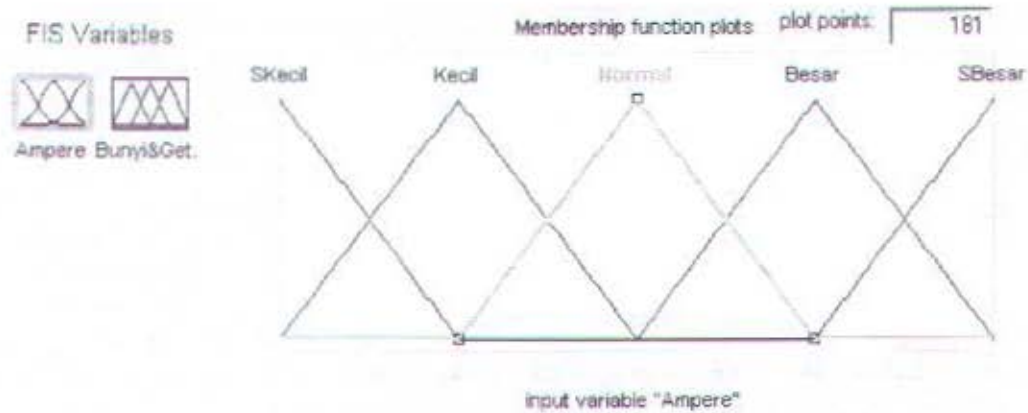


Gambar 3.17 Fungsi keanggotaan kondisi tekanan keluar P4 oli.

➤ Inputan : Ampere pompa dengan Outputan : Bunyi dan Getaran

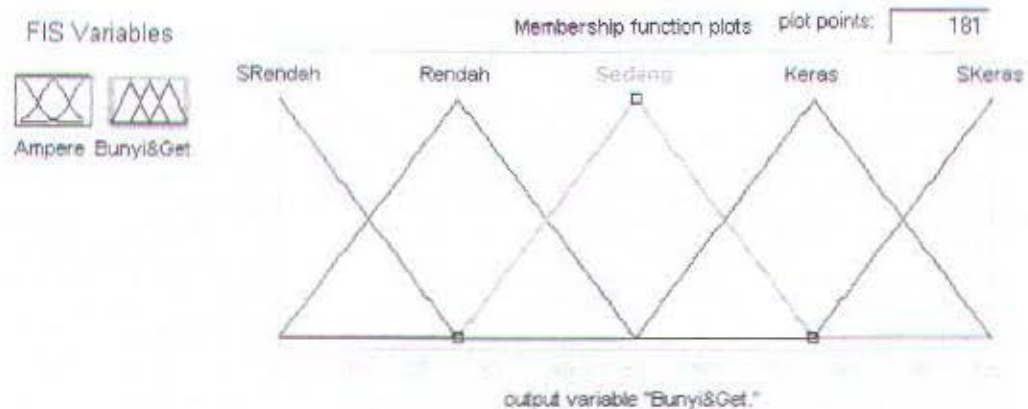


Input : Ampere pompa



Gambar 3.18 Fungsi keanggotaan ampere pompa oli.

Output : Bunyi dan Getaran (dalam x putaran pompa)



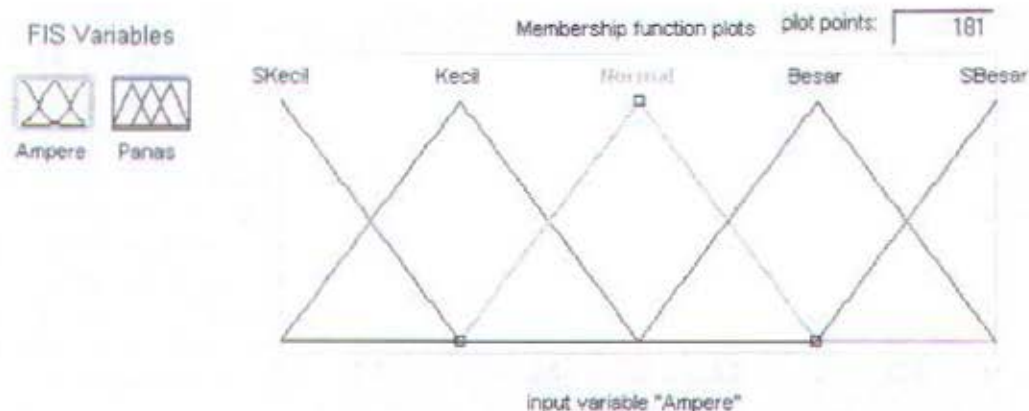
Gambar 3.19 Fungsi keanggotaan Bunyi & getaran pompa oli.

➤ Inputan : Ampere pompa dengan Outputan : Panas pompa



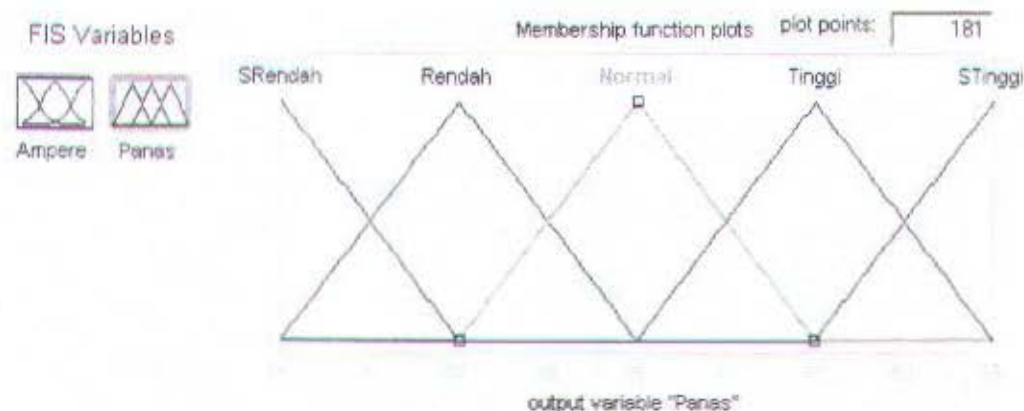


Input : Ampere pompa



Gambar 3.20 Fungsi keanggotaan ampere pompa oli.

Output : Panas pompa (dalam °C)

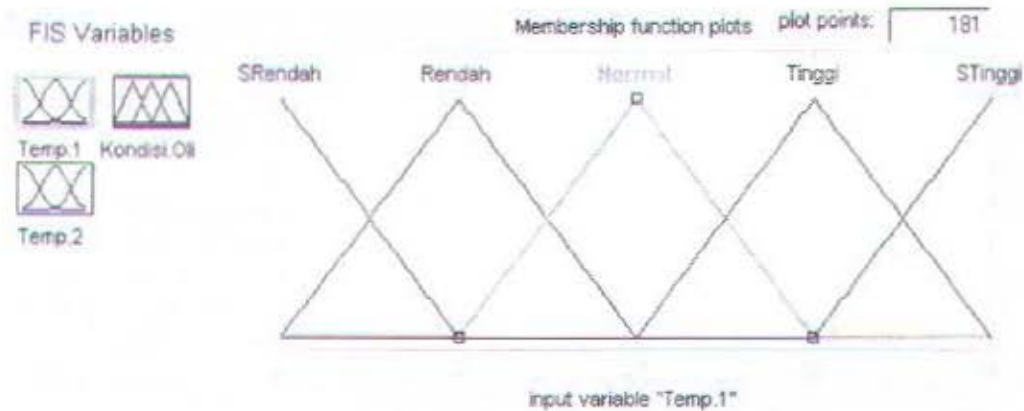


Gambar 3.21 Fungsi keanggotaan panas dari pompa oli.

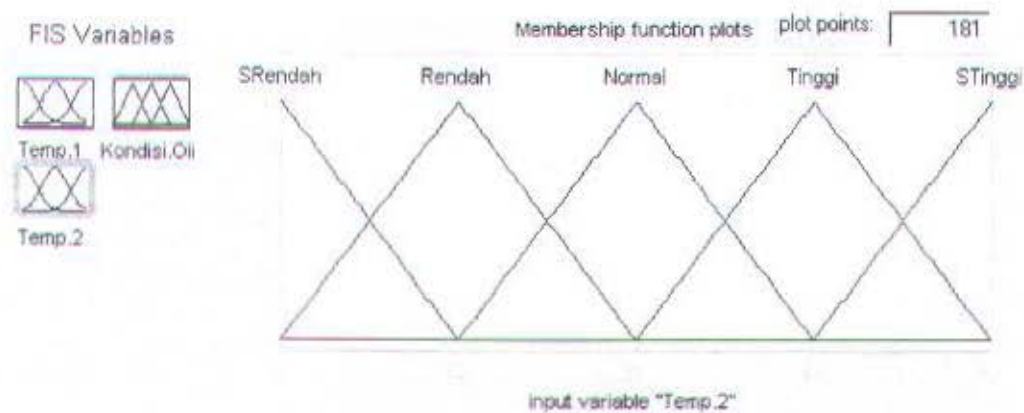
- Inputan : Temperatur oli sebelum masuk ke cooler / T1 (dalam °C) dan Temperatur oli setelah keluar dari cooler / T2 (dalam °C) dengan Outputan : Kondisi kekentalan dari oli.



Input : T1 (dalam °C) dan T2 (dalam °C)

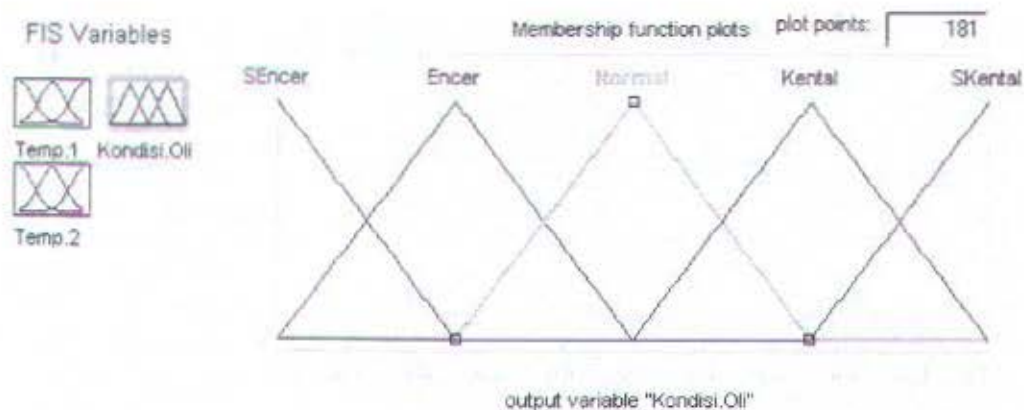


Gambar 3.22 Fungsi keanggotaan temperatur 1 oli.



Gambar 3.23 Fungsi keanggotaan temperatur 2 oli.

Output : Kondisi kekentalan dari oli



Gambar 3.24 Fungsi keanggotaan kondisi kekentalan oli.



3.5.4. Penyusunan Aturan Dasar

Kaidah-kaidah pengaturan *fuzzy* merupakan kumpulan aturan-aturan pengaturan sebagai acuan untuk menyatakan aksi pengaturan. Aturan-aturan tersebut disusun berdasarkan pengamatan atau perkiraan terhadap sistem yang berpedoman pada pengalaman-pengalaman si operator kapal dan juga *text book* tentang *trouble shooting*.

Dalam susunan aturan dasar ini digunakan sistem SISO dan DISO.

Aturan-aturan tersebut, yakni :

➤ Untuk sistem pelumas :

1. Inputan : Volume oli pada sump tank dan ouputan : lakukan pengisian



Gambar 3.25 Aturan dari kejadian input dan ouput pertama dari sistem oli.

2. Inputan : Switch pompa dan rpm Outputan : Kondisi pompa

- Jika Switch pompa stand by On dan Rpm 0 Maka Pompa tidak normal,
Juga
- Jika Switch pompa stand by On dan $Rpm \leq 1150$ Maka Pompa
berjalan normal, Juga
- Jika Switch pompa stand by Off dan Rpm 0 Maka Pompa mati



3. Inputan : Tekanan hisap dan tekanan hisap kemasukan udara (P1) dengan

Outputan : Tekanan keluar (P2)

```
1. If (Tek.Hisap is SRendah) and (Tek.H.K.U. is SBesar) then (Tek.Keluar is SKecil) (1)
2. If (Tek.Hisap is SRendah) and (Tek.H.K.U. is Besar) then (Tek.Keluar is SKecil) (1)
3. If (Tek.Hisap is SRendah) and (Tek.H.K.U. is Sedang) then (Tek.Keluar is Kecil) (1)
4. If (Tek.Hisap is SRendah) and (Tek.H.K.U. is Kecil) then (Tek.Keluar is Kecil) (1)
5. If (Tek.Hisap is SRendah) and (Tek.H.K.U. is SKecil) then (Tek.Keluar is Kecil) (1)
6. If (Tek.Hisap is Rendah) then (Tek.Keluar is Kecil) (1)
7. If (Tek.Hisap is Normal) then (Tek.Keluar is Normal) (1)
8. If (Tek.Hisap is Tinggi) then (Tek.Keluar is Besar) (1)
9. If (Tek.Hisap is STinggi) then (Tek.Keluar is SBesar) (1)
```

Gambar 3.26 Aturan dari kejadian input dan ouput ketiga dari sistem oli.

4. Inputan : Ampere pompa dengan Outputan : Bunyi dan Getaran dan Panas

dari pompa oli

```
1. If (Ampere is SKecil) then (Bunyi&Get. is SRendah) (1)
2. If (Ampere is Kecil) then (Bunyi&Get. is Rendah) (1)
3. If (Ampere is Normal) then (Bunyi&Get. is Sedang) (1)
4. If (Ampere is Besar) then (Bunyi&Get. is Keras) (1)
5. If (Ampere is SBesar) then (Bunyi&Get. is SKeras) (1)
```

(a)

```
1. If (Ampere is SKecil) then (Panas is SRendah) (1)
2. If (Ampere is Kecil) then (Panas is Rendah) (1)
3. If (Ampere is Normal) then (Panas is Normal) (1)
4. If (Ampere is Besar) then (Panas is Tinggi) (1)
5. If (Ampere is SBesar) then (Panas is STinggi) (1)
```

(b)

Gambar 3.27 Aturan dari kejadian input dan ouput keempat dari sistem oli
(a) untuk ouput bunyi dan getaran dari pompa, (b) untuk panas dari pompa.



5. Inputan : Perubahan tekanan keluar setelah melewati filter (ΔP_3) dengan

Outputan : Kondisi tekanan keluar setelah melewati filter (P_3)

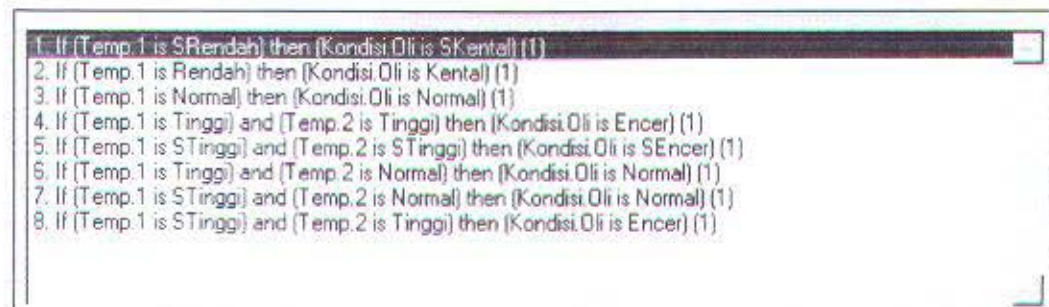


Gambar 3.28 Aturan dari kejadian input dan ouput kelima dari sistem oli

6. Inputan : Temperatur oli sebelum masuk ke cooler / T_1 (dalam $^{\circ}\text{C}$) dan

Temperatur oli setelah keluar dari cooler / T_2 (dalam $^{\circ}\text{C}$) dengan Outputan :

Kondisi kekentalan dari oli.



Gambar 3.29 Aturan dari kejadian input dan ouput ketujuh dari sistem oli

7. Inputan : Perubahan tekanan keluar setelah atau tidak melewati cooler (ΔP_4)

dengan Outputan : Kondisi tekanan keluar setelah atau tidak melewati cooler

(P_4)



Gambar 3.30 Aturan dari kejadian input dan ouput keenam dari sistem oli

3.5.5. Logika Pengambilan Keputusan

Pengambilan keputusan pada pengatur *fuzzy* adalah suatu prosedur untuk menghitung sinyal atur berdasarkan basis pengetahuan (basis data dan aturan dasar) yang sudah disusun. Prosedur ini berupa operasi matematik yaitu operasi mekanisme inferensi mamdani terhadap inputan yang diambil dalam pembahasan di sini.

3.5.6. Defuzzifikasi

Defuzzifikasi adalah merupakan teknik pengkonversian dari data *fuzzy* menjadi data *crisp*. Data *crisp* inilah yang akan diumpankan pada kontroler. Metode defuzzifikasi yang akan digunakan adalah COA (*Centre of Area*) atau juga disebut sebagai COG (*Centre of Gravity*), secara diskrit dinyatakan :

$$\Delta U = \frac{\sum_{j=1}^n \mu_z(w_j) w_j}{\sum_{j=1}^n \mu_z(w_j)}$$



di mana ΔU , $\mu_u(w_j)$ masing-masing adalah sinyal keluaran unit pengendali dan fungsi keanggotaan aksi pengaturan pada penyokong ke- w , sedangkan w_j adalah nilai tengah variabel linguistik bentuk segitiga aksi pengaturan.



B A B IV

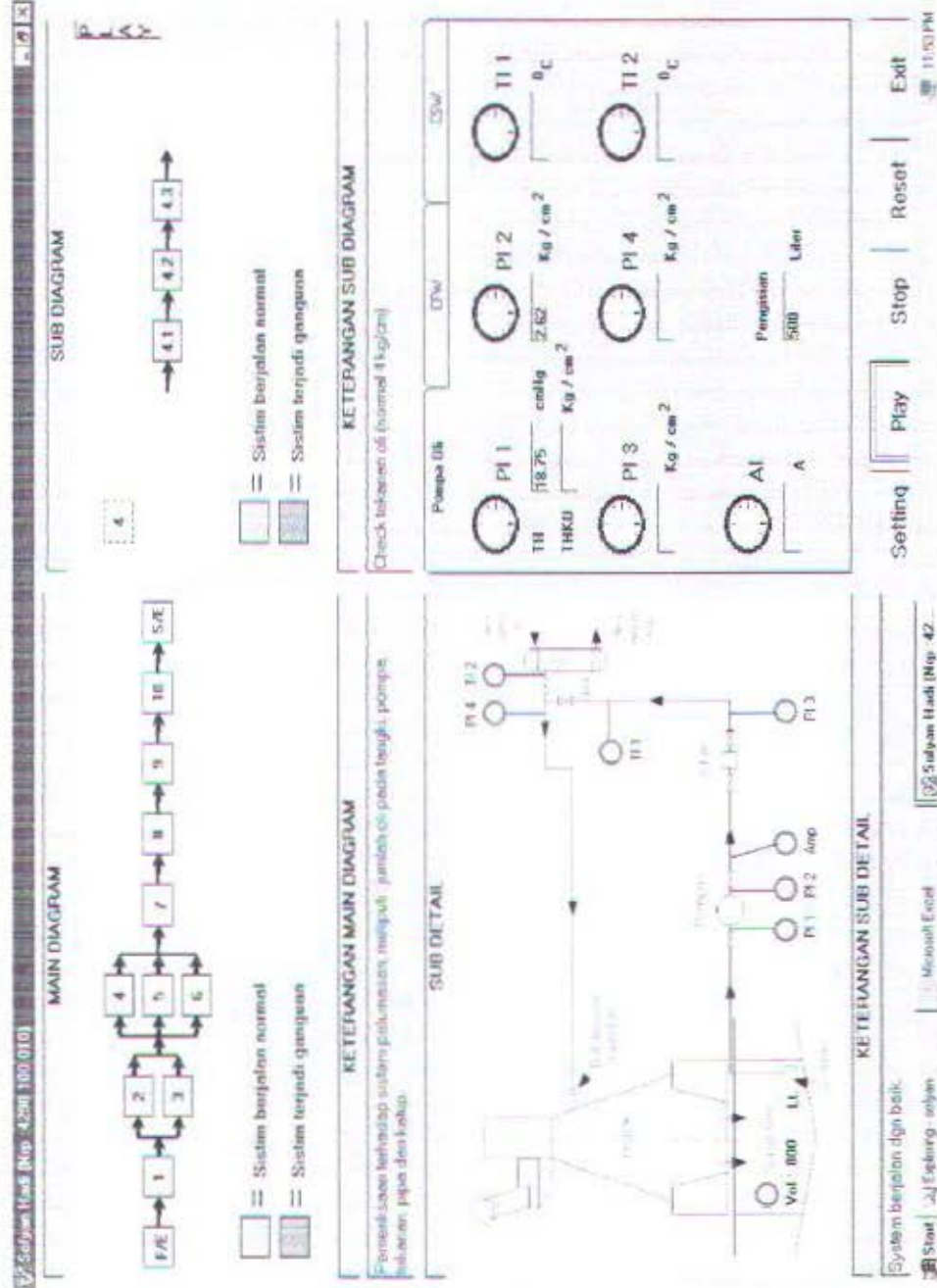
ANALISA DAN PEMBAHASAN

4.1. Simulasi

Dalam hal ini penyusunan fungsi keanggotaan *fuzzy* dari masing-masing inputan dan outputan di sini serta penyusunan aturan dasarnya dan mekanisme inferensinya menggunakan bantuan *software* Matlab versi 5.3.1. (Release 11.1). Sedangkan untuk tampilan menu utama pada *windows interfacenya* menggunakan bantuan *software* program Visual Basic Versi 6.0. Dikarenakan penyusun kesulitan dalam mengintegrasikan kedua *software* tersebut diatas dan juga kendala masalah waktu pembuatan programnya, maka untuk hasil dari input dan output *fuzzy* pada matlab dibuatkan dalam bentuk database pada program visual basic. Untuk hasil penggunaan *software* matlab terhadap input – output *fuzzy* dapat dilihat pada lampiran.

4.1.1. Tampilan Menu Utama

Pada simulasi di sini, telah dibuat juga tampilan-tampilan *windows* yang berkaitan dengan database dari data yang dibuat. Untuk lebih jelasnya, sebagai tampilan pembuka (menu utama) dapat dilihat pada gambar 4.1. Pada gambar ini nantinya adalah sebagai menu utama dari *windows interface*. Di sini akan terdapat beberapa bagian yang menjelaskan jalannya proses dari program. Di mana pada bagian-bagian tersebut ada yang berkaitan antara satu dengan yang lainnya.



Gambar 4.1 Tampilan menu utama.



Bagian-bagian pada tampilan menu utama ini terdiri dari *window* untuk *Main Diagram*, *window* untuk *Keterangan Main Diagram*, *window* untuk *Sub Diagram*, *window* untuk *Keterangan Sub Diagram*, *window* untuk *Sub Detail*, *window* untuk *Keterangan Sub Detail*, *window* untuk *output* serta tombol *Setting*, *Play*, *Stop*, *Reset* dan *Exit*. Lebih jelasnya dapat diketahui sebagai berikut :

Window Main Diagram

Pada *window* ini akan menampilkan mengenai diagram blok dari proses tahapan persiapan sebelum dilakukan *starting* motor induk. Di mana setiap blok ditandai dengan angka-angka yang diawali dengan blok F/E (*Finish Engine*) dan diakhiri dengan blok S/E (*Start Engine*). Angka-angka tersebut mulai dari angka 1 sampai dengan angka 10. Pada *window* ini pengoperasiannya ditandai dengan adanya warna biru muda yang menandakan tahapan tersebut sedang berjalan pada setiap bloknnya. Jika pada blok tersebut muncul warna hijau maka menandakan blok tersebut sudah dilalui dan dalam keadaan baik atau berjalan dengan normal. Sedangkan jika terjadi perubahan warna menjadi merah pada blok tersebut maka menandakan adanya atau terjadinya masalah atau *trouble*. Bersamaan dengan itu pada *window* yang terpisah (*window Keterangan Main Diagram*) akan muncul informasi mengenai blok tersebut.

Window Keterangan Main Diagram

Seperti yang diutarakan di atas bahwa *window* ini akan menampilkan informasi dari *window Main Diagram*. Informasi-informasi tersebut berupa hal-hal yang akan dilakukan dalam proses tahapan tersebut. Informasi ini hanya



muncul untuk satu fungsi dari setiap blok yang akan dilaluinya, jadi tidak secara keseluruhan dan bersamaan.

Window Sub Diagram

Pada *window* ini juga menampilkan mengenai diagram blok dari proses tahapan persiapan sebelum dilakukan *starting* motor induk. Hanya saja diagram blok yang muncul tersebut adalah blok-blok diagram yang ada di dalam blok pada *window main diagram* sewaktu kursor diarahkan pada setiap blok dari *window main diagram*. Jadi, dengan kata lain diagram blok ini akan muncul jika proses jalannya tahapan melalui pada setiap blok pada *window main diagram*. Namun, dalam hal ini tidak semua diagram blok ada pada setiap blok dari *window main diagram* akan muncul pada *window* ini. Pada *window* ini pengoperasiannya juga sama dengan *window main diagram*, yakni ditandai dengan adanya warna biru muda yang menandakan tahapan tersebut sedang berjalan pada setiap bloknya. Jika pada blok tersebut muncul warna hijau maka menandakan blok tersebut sudah dilalui dan dalam keadaan baik atau berjalan dengan normal. Sedangkan jika terjadi perubahan warna menjadi merah pada blok tersebut maka menandakan adanya atau terjadinya masalah atau *trouble*. Bersamaan dengan itu juga pada *window* yang terpisah (*window Keterangan Sub Diagram*) akan muncul informasi mengenai blok tersebut.

Window Keterangan Sub Diagram

Window ini juga menampilkan informasi dari *window Sub Diagram*. Informasi-informasi tersebut berupa hal-hal yang akan dilakukan dalam setiap bagian dari blok pada *window main diagram*. Informasi ini juga hanya muncul



untuk satu fungsi dari setiap blok yang akan dilaluinya, jadi tidak secara keseluruhan dan bersamaan.

Window Sub Detail

Pada *window* ini akan ditampilkan suatu skema digaram dari setiap sistem pada tahapan-tahapan yang dijalani. Namun, dalam pembahasan ini pada *window* ini hanya ditampilkan skema diagram sistem untuk sistem pelumasan dan sistem pendingin motor induk. Dikarenakan juga masalah kendala waktu maka hanya sistem pelumasan yang akan dibuatkan program proses pengaturan tahapan di dalamnya.

Window Keterangan Sub Detail

Pada *window* ini pada prinsipnya mempunyai fungsi yang sama dengan *window* keterangan-keterangan sebelumnya, yakni untuk memberikan atau menampilkan informasi-informasi yang ditujukan pada si operator. Hanya saja pada *window* ini informasi yang dimunculkan adalah mengenai proses tahapan yang akan dilalui pada setiap sistem, khususnya yang berhubungan dengan skema diagram dari sistem. Di sini juga ditandai dengan adanya beberapa indikasi yang digunakan sebagai arahan jalannya tahapan tersebut, apakah nantinya berjalan normal atau terjadi *trouble*. Indikasi-indikasi tersebut ditunjukkan berupa tanda lingkaran dalam *window* ini, yakni : Volume, tekanan (P), ampere (A) dan temperatur (T). Jika pada aliran sistem tersebut prosesnya berjalan normal maka semua bulatan-bulatan indikasi tersebut akan menunjukkan warna hijau, sedangkan jika bulatan tersebut menunjukkan warna merah maka pada bulatan yang berwarna merah tersebut terjadi atau mengalami *trouble*.

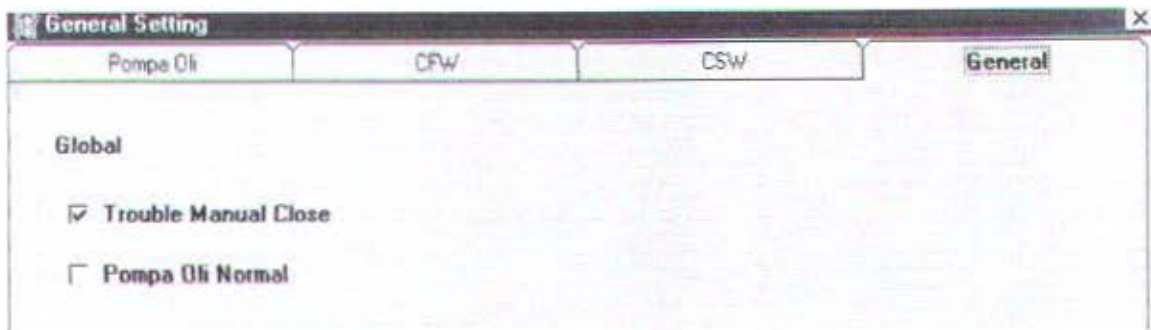


Window Output

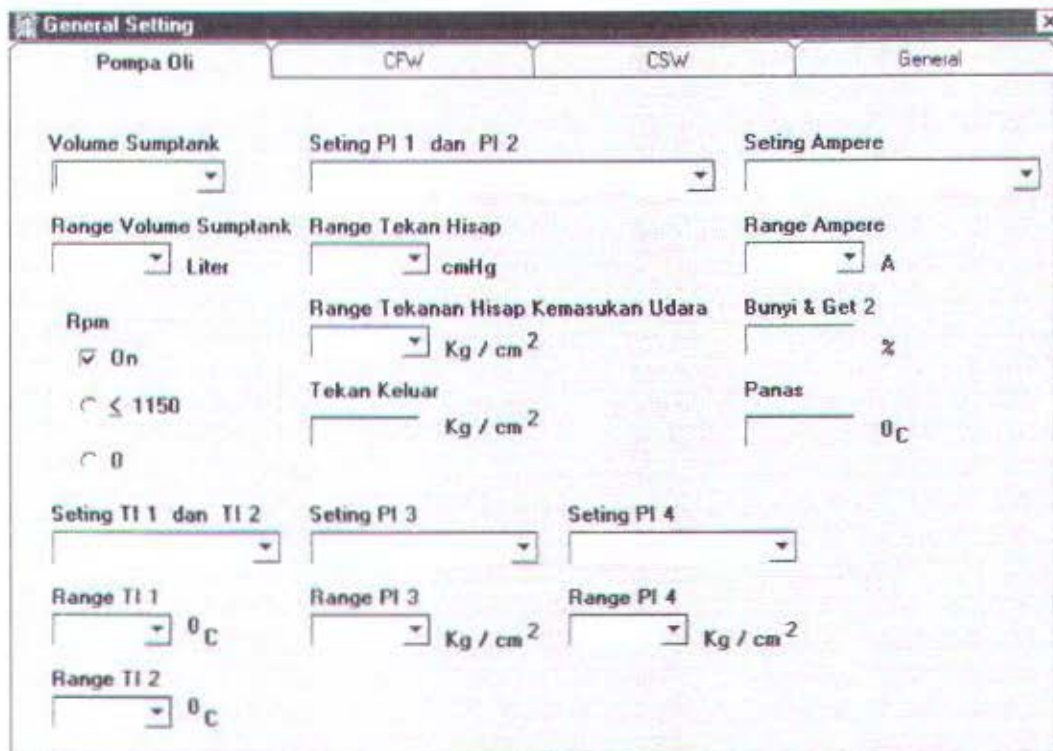
Untuk *window* ini, akan ditampilkan semua output yang ada. Sehingga pada *window* ini akan terdiri dari tiga *window* lagi di dalamnya, yakni : *window output* untuk sistem pelumasan, *window output* untuk sistem pendingin air tawar dan *window output* untuk sistem pendingin air laut. Dikarenakan yang dibahas hanya pada sistem pelumasan untuk pembuatan program pengaturan tahapan di dalam skema aliran sistemnya maka yang muncul pada *window* ini hanya outputan untuk sistem pelumasan saja. Masing-masing outputan tersebut adalah sudah berdasarkan hasil *setting* (untuk *window setting* akan dijelaskan pada bagian tersendiri) awal sebelum program ini dijalankan. Yakni berupa kotak tampilan hasil perhitungan *fuzzy* dan dibantu perhitungannya dalam matlab di mana dibuat dalam bentuk *database* pada program tersebut. Sehingga jika dilakukan pengisian *window setting* dan kemudian dilakukan penekanan tombol *play* pada *menu utama* maka pada *window output* akan muncul hasil dari perhitungan *fuzzy* tersebut. Dalam hal ini untuk sistem pelumas, baik itu berupa perintah untuk pengisian volume sump tank, tekanan, ampere dan temperatur.

Tombol Setting

Pada tombol ini mempunyai fungsi untuk menyetting awal dari program pengaturan tahapan-tahapan ini. Proses *setting* awal ini ditujukan pada sistem yang akan dilalui dan akan ditampilkan suatu tampilan *window* untuk *setting* awal. Dikarenakan dalam pembahasan di sini hanya sistem pelumasan yang dibuat, maka untuk *setting* awal ini yang ada pada *window setting* juga hanya untuk sistem pelumasan saja. Untuk lebih jelasnya tampilan *window setting* dapat dilihat pada gambar 4.2.



Gambar 4.2 Tampilan window general setting.



Gambar 4.3 Tampilan window setting sistem pelumasan.

Window yang muncul pada gambar 4.2 terdapat opsi dua pilihan, yakni apakah digunakan waktu penjalanan program tahapan ini memilih opsi yang pertama : *Trouble manual close* atau digunakan saat semua kondisi dari sistem tersebut (dalam hal ini sistem pelumasan) ingin berjalan normal, yakni memilih



opsi : *Pompa oli normal*. Di samping itu juga bisa dipergunakan pemilihan kedua opsi tersebut untuk proses kemudahan dan proses penyettingan yang lebih cepat.

Sedangkan *window* pada gambar 4.3 adalah tampilan untuk langkah selanjutnya setelah dilakukan penyettingan pada *window general setting*. Jika pada *window general setting* kita pilih opsi kedua maka pada *window* ini akan muncul data-data inputan *setting* yang khusus hanya sistem tersebut berjalan dengan normal. Sedangkan jika dipilih opsi yang pertama maka proses penyettingannya akan dilakukan satu per satu, yakni sesuai dengan skenario yang diinginkan. Seperti yang diuraikan sebelumnya yakni kedua opsi pada *general setting* dapat dipilih secara bersamaan, dengan tujuan jika diinginkan proses penyettingan yang lebih cepat dan jika di dalam skenario tersebut diinginkan ada proses tahapan yang mengalami *trouble*. Setelah proses penyettingan selesai dilakukan maka untuk keluar dari *window* ini cukup menekan tombol silang di ujung atas paling kanan.

Tombol Play

Tombol ini digunakan untuk menjalankan program yang ada. Artinya setelah semua proses *setting* awal telah dilakukan maka langkah selanjutnya adalah menjalankan program pengaturan tahapan tersebut. Yakni dengan cara menekan tombol *play* pada *window menu utama*.

Tombol Stop

Tombol ini mempunyai fungsi yang berkebalikan dengan fungsi tombol *play*, yakni tombol yang digunakan untuk menghentikan jalannya program.



Tombol Reset dan Exit

Fungsi dari tombol *reset* di sini adalah untuk menghapus atau mengganti data inputan yang ada pada *window setting*. Yakni dengan cara menekan tombol tersebut sekali maka semua data masukan yang sudah terisi pada *window setting* tersebut akan terhapus secara keseluruhan. Sedangkan untuk tombol *exit*, digunakan jikalau ingin keluar dari program tersebut.

4.1.2. Tampilan Jikalau Terdapat Trouble

Pada bagian ini akan ditunjukkan window yang menjelaskan setiap data masukan pada *window input*. Misalkan saja diambil contoh kondisi pada bagian sistem pelumasan terjadi ketidak normalan yaitu dengan *setting* awal Ampere Besar maka Panas Tinggi. Sewaktu proses tahapan berjalan dan sampai pada bagian yang dimaksud, untuk mengetahui adanya *trouble* akan ditunjukkan warna dari bulatan Ampere pada *window sub detail* menjadi warna merah. Di samping itu juga pada monitor akan muncul tampilan untuk window trouble pada kejadian ini seperti yang terlihat dalam gambar 4.4. *Trouble* yang terjadi pun pada *window* ini akan diketahui letaknya, apakah di daerah suction, system maupun mechanical. Dalam window ini pun juga akan ditunjukkan aksi yang akan dilakukan oleh si operator kapal nantinya. Pada gambar ini merupakan tampilan awal dari *window trouble*. Pada tampilan tersebut terdiri dua *windows* dan beberapa tombol yang mempunyai kaitan dengan *windows* tersebut. Dua *windows* tersebut adalah *window* untuk aksi yang akan dilakukan oleh si operator kapal terhadap *trouble* yang ada dan *window* keterangan yang menjelaskan informasi dari setiap aksi yang ada.



Sedangkan fungsi dari tombol-tombol yang berada di bawah seperti : tombol *suction*, tombol *system* dan tombol *mechanical* adalah merupakan tombol yang berfungsi sebagai petunjuk tentang adanya *trouble* yakni dengan ditunjukkan dalam munculnya warna merah pada tombol tersebut dan sekaligus juga untuk mengetahui aksi apa yang akan dilakukan oleh si operator kapal beserta keterangannya. Untuk mengetahui informasi pada *window* aksi dan *window* keterangan maka diantara ketiga tombol atau ketiga-tiganya menandakan warna merah pada tombol tersebut harus dilakukan penekanan terhadap tombol itu atau dengan cara diklik. Sehingga akan muncul seperti pada gambar 4.5.

Pada gambar 4.5 tersebut pada *window* aksi dan *window* keterangannya akan muncul informasi yang akan disampaikan pada operator kapal. Jika aksi yang dilakukan pada *trouble* tersebut lebih dari satu maka tombol *next* pada *window* aksi akan menjadi aktif dan dapat digunakan untuk melihat aksi berikutnya dengan cara menekan tombol tersebut. Penekanan tombol *next* akan mengaktifkan juga tombol *back*, yaitu untuk mengetahui aksi sebelumnya. Sedangkan untuk tombol *exit* yang berada di bagian pojok bawah sebelah kanan berfungsi untuk keluar dari *window trouble*.

Gambar 4.4 dan gambar 4.5 adalah contoh *trouble* dari salah satu output terhadap input saat *setting* awal berupa ampere. Sedangkan jika untuk *trouble-trouble* yang lain, pada dasarnya tampilan awalnya adalah sama. Hanya saja untuk



tanda adanya *trouble* diantara ketiga tombol berikut, yakni : *Suction*, *System* dan *Mechanical* kemungkinan tidak sama.

| TROUBLES | |
|------------|--|
| AKSI | |
| Keterangan | |

SUCTION SYSTEM MECHANICAL EXIT

Gambar 4.4 Tampilan awal window trouble untuk output panas dengan input ampere pada sistem pelumas.

| TROUBLES | |
|--------------|---|
| AKSI Next | 1. Check Katup relief |
| Keterangan | Kemungkinan pengesetannya terlalu rendah, periksa dan Re-set kembali. |

SUCTION SYSTEM MECHANICAL EXIT

Gambar 4.5 Tampilan window trouble untuk output panas dengan input ampere pada sistem pelumas.



Selain digunakan untuk menunjukkan adanya *trouble* dari setiap proses tahapan yang dilakukan (khususnya sistem pelumas), pada *window* ini juga nantinya dapat ditunjukkan pula mengenai beberapa informasi yang berhubungan dengan proses tahapan tersebut berjalan dengan normal atau baik. Maksudnya jika tahapan-tahapan dari sistem pelumas dalam hal ini berjalan normal atau pada saat settingnya diinginkan untuk berjalan normal secara keseluruhan, maka untuk memberikan informasi kepada si operator kapal bahwa sistem tersebut masih bekerja atau berjalan dengan normal adalah melalui *window* ini juga. Yaitu dengan memunculkan informasi-informasi tersebut pada bagian *window* keterangan. Namun saat kemunculan informasi saat kejadian tahapan tersebut berjalan normal ketiga tombol sumber daerah keberadaan *trouble* (suction, system dan mechanical) tidak dalam keadaan aktif.

Jilka dalam penyettingan awal ada salah satu tahapan yang dibuat tidak dalam kondisi normal sehingga memunculkan adanya *trouble* dan membuat indikasi bulatan pada *window sub detail* (dalam hal ini skema aliran sistem pelumas) menjadi warna merah, maka proses tahapan tersebut menjadi berhenti sementara. Jika diinginkan untuk melanjutkan kembali, maka kejadian dari tahapan yang mengalami ketidak normalan itu harus dibuat menjadi normal kembali. Dalam pengertian harus dilakukan perbaikan, seperti jikalau hal itu terjadi di kapal maka operator kapal harus membuat hal itu menjadi normal atau berjalan dengan baik kembali. Untuk menjadikan demikian pada program ini, maka setelah *window trouble* muncul dan kemudian ditutup dengan menekan tombol *exit* maka dilakukan penyettingan kembali pada bagian proses tahapan



yang mengalami *trouble* atau ketidak normalan. Sehingga hal itu nantinya akan menjadikan indikasi bulatan pada *window sub detail* yang tidak normal tersebut (dalam keadaan warna merah) akan menunjukkan warna hijau dan kemudian proses tahapan berlanjut kembali.

4.2. Analisa dan Pembahasan Sistem

Dari hasil program yang di tampilkan, ada beberapa kelemahan yang didapatkan. Diantaranya adalah tidak semua proses tahapan yang ada pada *window main diagram* dibuatkan program pengaturannya seperti yang dilakukan pada sistem pelumasan. Hal ini dikarenakan masalah kendala waktu yang tidak memungkinkan bagi penulis untuk membuat kesemuanya. Sehingga dalam hal ini jika proses tahapan dari setiap blok pada *window main diagram* berjalan (selain sistem pelumas) penulis menganggap proses tersebut sudah dalam keadaan normal. Jadi tidak ada proses penyettingan seperti yang dilakukan pada sistem pelumasan.

Kemudian untuk menampilkan informasi dari proses tahapan yang ada pada sistem itu sendiri yaitu jikalau proses tahapan dari sistem tersebut berjalan normal, tampilannya diambil dari *window trouble* jadi tidak memiliki *window* tersendiri. Meskipun sebenarnya hal itu masih bisa diatasi yaitu dengan tidak mengaktifkan tombol-tombol daerah terjadinya *trouble*. Selain itu juga mengenai tampilan outputnya pada setiap proses tahapan sistem yang ada. Dikarenakan di sini hanya sistem pelumasan yang diambil maka untuk sistem pendingin dan yang lain tidak buat. Tampilan outputnya di sini ditunjukkan berupa kotak keluaran



yang berisi angka dari hasil perhitungan *fuzzy* sebelumnya dan disertai juga gambar berupa petunjuk jarum. Namun gambar tersebut hanya diam dan tidak bisa berjalan sendiri, dikarenakan pada gambar tersebut tidak dibuatkan program juga masalah kendala waktu pembuatannya.

Perhitungan *fuzzy* dalam pembahasan di sini penulis mengambil bantuan *software* matlab. Dikarenakan penulis tidak bisa menggabungkan antara *software* matlab dengan *software* program *visual basic* maka penulis membuat hasil perhitungan *fuzzy* dibuatkan dalam bentuk *data base* pada program *visual basic*.

Di samping kelemahan-kelemahan tersebut hasil dari program ini juga memiliki kelebihan jikalau proses pengaturan tersebut hanya dilakukan dengan cara konvensional (yakni harus memeriksa satu-persatu dan meninjau langsung). Dengan dibuatkannya program pengaturan tahapan-tahapan dari proses jalannya persiapan sebelum motor induk dilakukan *starting*, maka dalam hal ini akan menjadi lebih mudah untuk dilakukan pengamatannya oleh operator kapal jika seandainya hal tersebut nantinya diterapkan pada lapangan.

Serta jikalau hal ini diterapkan pada kapal maka akan menjadikan suatu pengaturan pada proses tersebut menjadi lebih mudah bagi si operator kapal terutama bagi kru-kru kapal baru yang menggantikan kru-kru kapal yang lama. Sehingga program ini dapat membuat mereka akan lebih cepat memahami karakteristik dari proses tahapan-tahapan motor induk tersebut. Oleh karena itu, dengan adanya hal ini maka proses pengenalan bagi kru-kru baru tersebut tidak



memerlukan adaptasi yang lama. Dikarenakan semua pengalaman si operator yang senior dan karakteristik dari semua peralatan permesinan yang berhubungan dengan proses tahapan-tahapan persiapan sebelum dilakukannya *starting* sudah dibuatkan dalam bentuk *database* yang diadopsi dari aturan-aturan *fuzzy* dan dibuat menjadi terstruktur rapih dalam komputer dengan tampilan lewat layar monitor sebagai *monitoring* kru kapal tersebut.

Dengan demikian dalam pengawasan mengenai proses tahapan-tahapan motor induk sebelum dilakukannya *starting* sampai saat dilakukan start, tidak membutuhkan kru-kru yang lebih banyak untuk melakukannya. Melainkan cukup satu atau dua orang yang bertugas. Sehingga dalam hal ini secara tidak langsung pada orientasi ke depan dalam merencanakan pembuatan kapal baru (khususnya kapal niaga) dengan ditunjang perkembangan teknologi otomatisasi yang begitu pesat dan maju maka pertimbangan penggunaan jumlah kru patut diperhitungkan.



B A B V

PENUTUP

5.1. Kesimpulan

- Logika *fuzzy* dalam pembahasan di sini, yakni pengaturan proses tahapan-tahapan persiapan *starting* motor induk (motor diesel), ternyata bisa diterapkan meskipun hasil yang dicapai belum optimal.
- Ketidak optimalan dari hasil di sini adalah dikarenakan dalam pembuatan aturan-aturan *fuzzy* yang didasarkan atas pengambilan input-output masih dirasa kurang sempurna. Yakni dalam hal jikalau dalam suatu sistem dari proses tahapan tersebut (sistem pelumas) berjalan atau bekerja dalam kondisi normal namun indikasi yang ada telah menunjukkan nilai dari batas minimum ataupun batas maksimum, program di sini tidak bisa memberikan semacam informasi atas hal apa yang akan terjadi dari kondisi tersebut di atas.
- Program pengaturan proses tahapan-tahapan persiapan *starting* motor induk (motor diesel) ini dapat memberikan kemudahan dalam aktivitas dari kru kapal untuk memonitor sehingga nantinya pertimbangan untuk memilih jumlah awak kapal dapat diminimalkan dan dapat memberikan kemudahan dalam memahami atau beradaptasi dengan karakteristik dari permesinan yang terkait bagi kru kapal *junior* atau *cadet* sehingga tidak memerlukan waktu adaptasi yang lama.



5.2. Saran

- Dalam pembahasan masalah di sini hanya diambil sistem pelumasan yang dibuat program pengaturannya lebih detail, untuk pengembangan lebih lanjut mungkin dapat dilanjutkan pada semua sistem yang ada dan jikalau memungkinkan juga dapat dilanjutkan untuk proses tahapan saat motor induk start sampai *running*. Sehingga diharapkan program yang dibuat tersebut nantinya akan menjadi lebih sempurna.
- Dibuat *interface* yang lebih *user friendly* sehingga lebih mempermudah hubungan antara operator dengan sistem yang dikembangkan.



DAFTAR PUSTAKA

- Arismunandar, Wiranto ; Koichi Tsuda [1993], **"Motor Diesel Putaran Tinggi"** ,
P.T. Pradnya Paramita, Cetakan Ketujuh, Jakarta.
- Budiyanto, Djoko [1995], **"Sistem Kontrol Berbasis Logika Fuzzy"** , Jurnal
Vasthu No. 03/Th. III, Edisi Juli.
- Chien Lee, Chuen [1990], **"Fuzzy Logic in Control System"** , Fuzzy Logic
Controller Part I, IEEE Transaction on System, Man and Cybernetics, Vol.
20 No. 2, Edisi Maret / April.
- Chien Lee, Chuen [1990], **"Fuzzy Logic in Control System"** , Fuzzy Logic
Controller Part II, IEEE Transaction on System, Man and Cybernetics, Vol.
20 No. 2, Edisi Maret / April.
- Harris, C. J. ; C. G. Moore ; M. Brown [1993], **"Intelligent Control"** , Aspects of
Fuzzy Logic and Neural Nets, World Scientific Publishing Co. Pte. Ltd.
- Jamshidi, M. ; Nader Vadiie ; Timothy J. Ross [1993], **"Fuzzy Logic and
Control : Software and Hardware Applications"** , Prentice Hall
International, Inc., New Jersey, U.S.A.
- Karassik, Igor J. ; William C. Krutzsch ; Warren H. Fraser ; Joseph P. Messina
[1976], **"Pump Handbook"** , McGraw – Hill Book Company, U.S.A.
- Maleev, V. L. ; Bambang Priambodo [1986], **"Operasi dan Pemeliharaan Mesin
Disel : Konstruksi, Operasi, Pemeliharaan dan Perbaikan Mesin Disel"**
, Penerbit Erlangga, Jakarta.



- Negnevitsky, Michael , **"Fuzzy Expert System"** , University of Tasmania.
- Patyra, M. J. [1996], **"Fuzzy Logic Implementation and Applications"** , Wiley
– Teubner.
- Sugeno, Michio [1991], **"Industrial Application of Fuzzy Control"** , Elseiver
Science Publishers.
- Suharini, Yustina Sri [1998], **"Hubungan Logika Kabur Dengan Basic Data Relasional (Studi Awal Fuzzy Database)"** , Jurnal Teknologi Industri Vol. II No. 3, Edisi Juli, F.T.I. – Universitas Atma Jaya, Yogyakarta.
- Tanujaya, Harlianto [1999], **"Penerapan Logika Fuzzy Sebagai Pengaturan Keadaan Ruangan Secara Otomatis"** , Jurnal Emas Th. IX No. 16, Edisi Februari, F.T. – U.K.I.
- Toboldt, W. K. Bill [1983], **"Diesel Fundamentals, Service, Repair"** , The Goodheart – Willcox Company, Inc., South Holland, Illinois.
- Warring, R. H. [1984], **"Pumping Manual"** , 7th Edition, Trade and Technical Press Limited, Great Britain, England.
- Yan, Jun ; M. Ryan ; J. Power [1994], **"Using Fuzzy Logic"** , Prentice Hall International (U.K.), Limited London.



UCAPAN TERIMA KASIH

Atas terselesainya pembuatan laporan Tugas Akhir ini, penulis memberikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang terkait, baik secara langsung maupun tidak langsung. Tanpa jasa-jasa mereka semua, penulis yakin pembuatan laporan ini tidak akan berhasil dengan sukses dan terselesaikan dengan baik. Ucapan terima kasih penulis yang sangat besar ditujukan kepada :

- Bapak DR. Ir. Agoes A. Masroeri, M.Eng. selaku Ketua Jurusan Teknik Sistem Perkapalan – FTK – ITS Surabaya serta selaku dosen pembimbing pertama buat penulis yang selalu memberikan arahan dan bimbingan kepada penulis.
- Bapak Ir. Aries P. Hidayatullah selaku dosen pembimbing kedua yang juga tidak sedikit pula memberikan masukan dan wawasan kepada penulis.
- Orang tua penulis yang telah turut membantu dukungan moril maupun maupun materiil.
- Nahkoda Kapal Motor Penumpang Adhiswadharma III milik P.T. Adhiguna Citra Utama, Bapak Juni Iskandar, yang telah memberikan ijin untuk mengambil dan menggali informasi di kapal tersebut.
- Kepala Kamar Mesin K.M.P. Adhiswadharma III, Bapak Mulyono, yang juga memberikan ijin untuk masuk ke kamar mesin dan memberikan wawasan serta informasi yang dibutuhkan oleh penulis.



- Para kru-kru K.M.P. Adhiswadharma lain yang juga ikut membantu memberikan informasi dari setiap pertanyaan yang penulis ajukan.
- Temanku Dudi'96, yang juga turut membantu memberikan masukan kepada penulis di kala mengalami kesulitan dalam memecahkan persoalan.
- Temanku Djoko'98, yang P1-nya sama-sama mengalami nasib serupa dan sepenanggungan.
- Temanku Arief'98, meskipun kadang-kadang juga sering memberikan kritik bagi penulis, penulis menganggap hal itu sudah biasa dan itu memang baik dan dibutuhkan.
- Temanku Heru'98, meskipun tidak terlibat langsung asalkan dia selalu memberikan senyuman dan bertanya tentang khabar penulis, penulis merasa senang.
- Temanku Ilham Wijaya Kusuma Wardana yang turut membantu dalam proses pengerjaan Tugas Akhir ini.
- Dan semua orang lain yang tidak bisa penulis sebutkan satu-persatu atas bantuannya selama dalam proses pengerjaan Tugas Akhir ini.

Berkat dukungan baik do'a maupun jasa merekalah pengerjaan Tugas Akhir ini bisa terselesaikan. Semoga amal ibadahnya akan dibalas oleh Allah SWT di lain hari kelak.

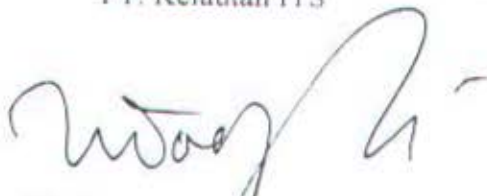


SURAT KEPUTUSAN Pengerjaan Tugas Akhir KS 1701


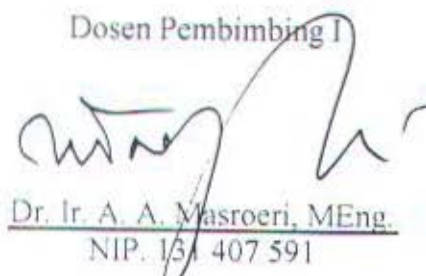

Sebagai salah satu syarat untuk mendapatkan gelar Sarjana Teknik Sistem Perkapalan Fakultas Teknologi Kelautan ITS, maka perlu ditertibkan Surat Keputusan Pengerjaan Tugas Akhir yang memberikan tugas kepada mahasiswa tersebut di bawah untuk mengerjakan Tugas sesuai judul dan lingkup bahasan yang ditentukan.

| | |
|----------------------------|--|
| Nama Mahasiswa | : SOFYAN HADI |
| NRP. | : 4298 100 010 |
| Dosen Pembimbing | : 1. Dr. Ir. Agoes A. Masroeri, MEng. 2. Ir. Aries P. Hidayatullah |
| Tanggal Diberikan Tugas | : Oktober 2002 |
| Tanggal Diselesaikan Tugas | : |
| Judul Tugas Akhir | : STUDI PENERAPAN LOGIKA FUZZY PADA SISTEM OPERASIONAL PERMESINAN KAPAL PENYEBERANGAN (FERRY) |

Surabaya, Oktober 2002
Ketua Jurusan Teknik Sistem Perkapalan
FT. Kelautan ITS


DR. Ir. Agoes A. Masroeri, M.Eng
NIP. 131 407 591

Surabaya, Oktober 2002
Yang menerima tugas:

| | | |
|---|---|--|
| Mahasiswa | Dosen Pembimbing I | Dosen Pembimbing II |
|  |  |  |
| <u>Sofyan Hadi</u> NRP. 4298 100 010 | <u>Dr. Ir. A. A. Masroeri, MEng.</u> NIP. 131 407 591 | <u>Ir. Aries P. Hidayatullah</u> NIP. 132 238 882 |

LIST PROGRAM UNTUK FORM MENU UTAMA

```
Private Const Jrk As Integer = 10
Dim fxi As Integer, sl As Integer
Dim fxa As Integer
Dim KetUtm(11), Ket4(2), Ket5(3), Ket6(1), Ket7(4), Ket8(1), OptScr As String

Sub PisahTmr()
    If sl >= 4 And sl <= 9 Then
        FrmSub(0).Visible = False
        Select Case sl
            Case Is = 4
                FrmSub(4).Visible = True
                FrmGlb(0).Visible = False: FrmGlb(4).Visible = True: FrmGlb(5).Visible =
False
                Timer1.Interval = 0: Timer1.Enabled = False
                Timer4.Interval = 200: Timer4.Enabled = True: NT = 4
            Case Is = 5
                FrmSub(4).Visible = False: FrmSub(5).Visible = True
                FrmGlb(0).Visible = False: FrmGlb(5).Visible = True
                Timer1.Interval = 0: Timer1.Enabled = False
                Timer5.Interval = 200: Timer5.Enabled = True: NT = 5
            Case Is = 6
                RichTextBox2.Text = Empty
                FrmSub(5).Visible = False: FrmSub(6).Visible = True
                FrmGlb(0).Visible = True
                Timer1.Interval = 0: Timer1.Enabled = False
                Timer6.Interval = 200: Timer6.Enabled = True: NT = 6
            Case Is = 7
                FrmSub(6).Visible = False: FrmSub(7).Visible = True
                Timer1.Interval = 0: Timer1.Enabled = False
                Timer7.Interval = 200: Timer7.Enabled = True: NT = 7
            Case Is = 8
                FrmSub(7).Visible = False: FrmSub(8).Visible = True
                Timer1.Interval = 0: Timer1.Enabled = False
                Timer8.Interval = 200: Timer8.Enabled = True: NT = 7
            Case Is = 9
                FrmSub(8).Visible = False
                FrmSub(0).Visible = True
                RichTextBox2.Text = Empty
        End Select
    End If
End Sub

Sub Lanjut()
    Select Case NT
        Case Is = 1
            Timer1.Interval = 200: Timer1.Enabled = True
        Case Is = 4
            Timer4.Interval = 200: Timer4.Enabled = True
```

```

Case Is = 5
    Timer5.Interval = 200: Timer5.Enabled = True
Case Is = 6
    Timer6.Interval = 200: Timer6.Enabled = True
Case Is = 7
    Timer7.Interval = 200: Timer7.Enabled = True
Case Is = 8
    Timer8.Interval = 200: Timer8.Enabled = True
End Select
End Sub
Sub Berhenti()
Select Case NT
    Case Is = 1
        Timer1.Interval = 0: Timer1.Enabled = False
    Case Is = 4
        Timer4.Interval = 0: Timer4.Enabled = False
    Case Is = 5
        Timer5.Interval = 0: Timer5.Enabled = False
    Case Is = 6
        Timer6.Interval = 0: Timer6.Enabled = False
    Case Is = 7
        Timer7.Interval = 0: Timer7.Enabled = False
    Case Is = 8
        Timer8.Interval = 0: Timer8.Enabled = False
End Select
End Sub

Sub Bersih()
For i = 0 To 11
    LblUtm(i).BackColor = vbWhite
    Select Case i
        Case Is = 0
            ImgVol(i).Visible = True: ImgPi1(i).Visible = True: ImgPi2(i).Visible = True
            ImgAmp(i).Visible = True: ImgTi1(i).Visible = True: ImgTi2(i).Visible = True
            ImgP1(i).Visible = True: ImgP2(i).Visible = True
            ImgP3(i).Visible = True: ImgP4(i).Visible = True
            FrmGlb(i).Visible = True: FrmSub(i).Visible = True
        Case 1 To 2
            ImgVol(i).Visible = False: ImgPi1(i).Visible = False: ImgPi2(i).Visible = False
            ImgAmp(i).Visible = False: ImgTi1(i).Visible = False: ImgTi2(i).Visible =
False
            ImgP1(i).Visible = False: ImgP2(i).Visible = False
            ImgP3(i).Visible = False: ImgP4(i).Visible = False
        Case 4 To 8
            FrmSub(i).Visible = False
        End Select
    End Select
Next i
For i = 1 To 5
    Label71(i).BackColor = vbWhite
    Select Case i

```

Case 1 To 2

Label41(i).BackColor = vbWhite

Label51(i).BackColor = vbWhite

Label52(i).BackColor = vbWhite

Label61(i).BackColor = vbWhite

Label81(i).BackColor = vbWhite

Case Is = 3

Label41(i).BackColor = vbWhite

ImgP3(i).Visible = False

ImgP4(i).Visible = False

Case 4 To 5

FrmGlb(i).Visible = False

End Select

Next i

NT = 1: fxa = 0: fxi = 0: sl = 0: sa = 1:

Jln = False: Strb = False

FrmSetting.Check1.Value = vbChecked

FrmSetting.Check2.Value = vbUnchecked

LblPlay.Visible = False: LblStop.Visible = False

LblTh.Caption = Empty: LblThKu.Caption = Empty

LblPI2.Caption = Empty: LblP3.Caption = Empty: LblP4.Caption = Empty

LblTI1.Caption = Empty: LblTI2.Caption = Empty

LblIsi.Caption = Empty: LblAmp.Caption = Empty

RichTextBox1.Text = Empty: RichTextBox2.Text = Empty: RichTextBox3.Text = Empty

End Sub

Sub Lihat()

Dim Benar As Boolean: Benar = False

With FrmTrouble

If SQL.Recordset.RecordCount > 0 Then

SQL.Recordset.MoveFirst

Do While Not SQL.Recordset.EOF

Select Case SQL.Recordset(2)

Case Is = "Suction": Timer1.Interval = 200: Timer1.Enabled = True

Case Is = "System": Timer2.Interval = 200: Timer2.Enabled = True

Case Is = "Mechanical": Timer3.Interval = 200: Timer3.Enabled = True

Case Is = "No"

.OptSct.Enabled = False

.OptSys.Enabled = False

.OptMch.Enabled = False

If IsNull(SQL.Recordset(3)) Then

.Text1.Text = Empty

Else

.Text1.Text = SQL.Recordset(3)

End If

.RichTextBox1.Text = SQL.Recordset(4)

End Select

SQL.Recordset.MoveNext

Loop


```
End If
.Show 1, Me
End With
End Sub
```

```
Private Sub CmdExit_Click()
End
End Sub
```

```
Private Sub CmdPlay_Click()
If Not Jln Then
MsgBox "aturlah general settingnya, baru bisa jalan", vbExclamation + vbOKOnly,
"Ok !!!"
CmdSet.SetFocus
Else
LblPlay.Visible = True
LblStop.Visible = False
Call Lanjut
End If
End Sub
```

```
Private Sub CmdReset_Click()
Call Berhenti
If MsgBox("Apakah anda yakin untuk di RESET", vbQuestion + vbYesNo, "Jangan
Ragu - Ragu") = vbYes Then
Timer1.Interval = 0: Timer4.Enabled = False
Timer4.Interval = 0: Timer4.Enabled = False
Timer5.Interval = 0: Timer5.Enabled = False
Timer6.Interval = 0: Timer6.Enabled = False
Timer7.Interval = 0: Timer7.Enabled = False
Timer8.Interval = 0: Timer8.Enabled = False
Call Bersih
Else
If LblPlay.Visible Then
Call Lanjut
End If
End If
End Sub
```

```
Private Sub CmdSet_Click()
FrmSetting.Show 0, Me
End Sub
```

```
Private Sub CmdStop_Click()
If Not LblStop.Visible And Not LblPlay.Visible Then
MsgBox "filmnya belum dijalankan, kok mau diberhentikan sih", vbInformation +
vbOKOnly, "Pesan Canda"
Exit Sub
End If
LblStop.Visible = True
```

```
LblPlay.Visible = False
Call Berhenti
End Sub
```

```
Private Sub Command1_Click()
    LblUtm(11).BackColor = vbGreen: NT = 1
```

```
End Sub
```

```
Private Sub Form_Activate()
    If Jln And SSet Then
        LblPlay.Visible = True
        LblStop.Visible = False
        Select Case NT
            Case Is = 1
                Timer1.Interval = 200: Timer1.Enabled = True
            Case Is = 4
                Timer4.Interval = 200: Timer4.Enabled = True
            Case Is = 5
                Timer5.Interval = 200: Timer5.Enabled = True
            Case Is = 6
                Timer6.Interval = 200: Timer6.Enabled = True
            Case Is = 7
                Timer7.Interval = 200: Timer7.Enabled = True
            Case Is = 8
                Timer8.Interval = 200: Timer8.Enabled = True
        End Select
    End If
End Sub
```

```
Private Sub Form_Load()
    NT = 1: sa = 1: Jln = False
    SQL.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
    App.Path & "\VBaru.mdb;Persist Security Info=False"
    SQL.CommandType = adCmdText
    For i = 4 To 8
        FrmSub(i).Top = FrmSub(0).Top: FrmSub(i).Left = FrmSub(0).Left:
        FrmSub(i).Visible = False
        If (i = 4 Or i = 5) Then
            FrmGlb(i).Top = FrmGlb(0).Top: FrmGlb(i).Left = FrmGlb(0).Left:
            FrmGlb(i).Visible = False
        End If
    Next i
    For i = 1 To 4
        If i = 4 Then
            ImgP3(i).Top = ImgP3(0).Top: ImgP3(i).Left = ImgP3(0).Left: ImgP3(i).Visible
            = False
            ImgP4(i).Top = ImgP4(0).Top: ImgP4(i).Left = ImgP4(0).Left: ImgP4(i).Visible
            = False
        Else
```



```

    ImgVol(i).Top = ImgVol(0).Top: ImgVol(i).Left = ImgVol(0).Left:
    ImgVol(i).Visible = False
    ImgPi1(i).Top = ImgPi1(0).Top: ImgPi1(i).Left = ImgPi1(0).Left:
    ImgPi1(i).Visible = False
    ImgPi2(i).Top = ImgPi2(0).Top: ImgPi2(i).Left = ImgPi2(0).Left:
    ImgPi2(i).Visible = False
    ImgAmp(i).Top = ImgAmp(0).Top: ImgAmp(i).Left = ImgAmp(0).Left:
    ImgAmp(i).Visible = False
    ImgTi1(i).Top = ImgTi1(0).Top: ImgTi1(i).Left = ImgTi1(0).Left:
    ImgTi1(i).Visible = False
    ImgTi2(i).Top = ImgTi2(0).Top: ImgTi2(i).Left = ImgTi2(0).Left:
    ImgTi2(i).Visible = False
    ImgP3(i).Top = ImgP3(0).Top: ImgP3(i).Left = ImgP3(0).Left: ImgP3(i).Visible
= False
    ImgP4(i).Top = ImgP4(0).Top: ImgP4(i).Left = ImgP4(0).Left: ImgP4(i).Visible
= False
    End If
Next i
'fxi = 0: sl = 0
KetUtm(0) = "Motor induk dalam keadaan berhenti atau mati."
KetUtm(1) = "Hidupkan blower untuk ventilasi di kamar mesin " & _
    "(blower harus selalu dihidupkan meskipun ME/off)."
KetUtm(2) = "Pemeriksaan terhadap bagian-bagian dari M/E yang bergerak " & _
    "(katup, nok, penggerak katup, pompa bahan bakar, sistem injeksi bahan " & _
    "bakar, governor) untuk penyetelan dan pelumasan yang baik."
KetUtm(3) = "Pemeriksaan terhadap bagian-bagian dari M/E dan permesinana " & _
    "yang tidak bergerak, meliputi mur, baut, sambungan, jaket."
KetUtm(4) = "Pemeriksaan terhadap sistem pelumasan, meliputi : jumlah oli pada " & _
    "tangki, pompa, tekanan, pipa dan katup."
KetUtm(5) = "Pemeriksaan terhadap sistem pendingin, meliputi : sistem pendingin "
& _
    "air tawar dan air laut." & Chr(10) & _
    "Pada sistem pendingin air tawar meliputi : jumlah level air pada " & _
    "tangki ekspansi, tekanan, pompa, pipa dan katup." & Chr(10) & _
    "Pada sistem pendingin air laut meliputi : saluran hisap, tekanan, pompa dan
katup."
KetUtm(6) = "Pemeriksaan terhadap sistem bahan bakar, yaitu meliputi pemeriksaan "
& _
    "jumlah bahan bakar pada tangki, pompa, pipa dan katup."
KetUtm(7) = "Pemeriksaan terhadap sistem udara start, meliputi : tekanan dalam " & _
    "tangki reservoir, pipa, katup start."
KetUtm(8) = "Pemeriksaan terhadap sistem udara bilasdan buang, meliputi air " & _
    "intake screen dan exhaust shut off valve."
KetUtm(9) = "Pemeriksaan terhadap beban M/E " & _
    "(harus diputuskan dan M/E tidak boleh dalam keadaan terbebani)."
KetUtm(10) = "Buka katup penstarter udara."
KetUtm(11) = "Proses starting motor induk."
Ket4(0) = "Check volume oli pada tangki."

```



```

Ket4(1) = "Jalankan motor pompa stand by."
Ket4(2) = "LO System."
Ket5(0) = "Check level volume air tawar pada tangki ekspansi."
Ket5(1) = "Buka saluran hisap Sea Water (katup air laut)."
Ket5(2) = "Jalankan motor pompa Fresh Water."
Ket5(3) = "Jalankan motor pompa Sea Water."
Ket6(0) = "Ceck penyediaan bahan bakar di dalam tangki."
Ket6(1) = "Kendali pompa bahan bakar pada kedudukan Fuel On."
Ket7(0) = "Check tekanan pada botol angin / reservoir (maksimum 29 kg/cm)."
Ket7(1) = "Lakukan pengisian jika terjadi kekurangan."
Ket7(2) = "Check tekanan kembali."
Ket7(3) = "Check apakah terdapat kebocoran / penyumbatan (pada pipa)."
Ket7(4) = "Check katub start."
Ket8(0) = "Check air intake screen."
Ket8(1) = "Buka katub exhaust shut off."
End Sub

```

```

Private Sub Timer1_Timer()
If fxi < (Jrk + (Jrk * sl)) Then
If sl <= 11 Then
If RichTextBox1.Text <> KetUtm(sl) Then
RichTextBox1.Text = KetUtm(sl)
LblUtm(sl).BackColor = 16776960
End If
End If
If sl = 13 Then
Call Berhenti
Call Bersih
End If
If sl > 0 Then
Call PisahTmr
LblUtm(sl - 1).BackColor = vbGreen
End If
End If
fxi = fxi + 1
sl = fxi \ Jrk
End Sub

```

```

Private Sub Timer4_Timer()
If fxa <= Jrk Then
If RichTextBox2.Text <> Ket4(0) Then
RichTextBox2.Text = Ket4(0)
End If
ImgVol(3).Visible = True
Label41(1).BackColor = 16776960
If fxa = Jrk Then
Call Buka(sa)
LblVol.Caption = FrmSetting.CbmRngSumpTank.Text
SQL.RecordSource = "SELECT CInt([A_Isi]*[VarVol]) AS VarVol, A_Isi.Isi
FROM A_Isi WHERE VarVol=" & CInt(LblVol.Caption)

```

```

SQL.Refresh: LblIsi.Caption = SQL.Recordset(1)
NCbm = (FrmSetting.CbmSumpTank.ListIndex + 1)
SQL.RecordSource = "select * from " & SetData & " where Kondisi=" & NCbm
SQL.Refresh
If Not IsNull(SQL.Recordset(0).Value) Then RichTextBox3.Text =
SQL.Recordset(0).Value
For i = 0 To 3
    ImgVol(i).Visible = False
Next i
If SQL.Recordset(1).Value = 3 Then
    ImgVol(2).Visible = True: Label41(1).BackColor = vbGreen
    sa = (sa + 1)
Else
    ImgVol(1).Visible = True: Label41(1).BackColor = vbRed
    fxa = (fxa - Jrk)
End If
FrmTrouble.Caption = SSTab1.TabCaption(0) & " Isi = " & LblIsi.Caption & "
liter"
Call Lihat: Timer4.Interval = 0: Timer4.Enabled = False
End If
ElseIf fxa <= (Jrk * 2) Then
    If RichTextBox2.Text <> Ket4(1) Then
        RichTextBox2.Text = Ket4(1)
    End If
    Label41(2).BackColor = 16776960
    If fxa = (Jrk * 2) Then
        Call Buka(sa)
        If FrmSetting.Check3 = vbUnchecked Then
            NCbm = 3: StrOpt = FrmSetting.OptRpm(1).Caption & " (dalam kondisi Off)."
        Else
            If FrmSetting.OptRpm(0).Value Then NCbm = 1: StrOpt =
FrmSetting.Label1A(31).Caption & " (dalam kondisi On)."
            If FrmSetting.OptRpm(1).Value Then NCbm = 2: StrOpt =
FrmSetting.OptRpm(1).Caption & " (dalam kondisi On)."
        End If
        SQL.RecordSource = "select * from " & SetData & " where Kondisi=" & NCbm
        SQL.Refresh
        If Not IsNull(SQL.Recordset(0).Value) Then RichTextBox3.Text =
SQL.Recordset(0).Value
        FrmTrouble.Caption = SSTab1.TabCaption(0) & " Rpm = " & StrOpt
        If NCbm = 1 Then
            Label41(2).BackColor = vbGreen
            sa = (sa + 1)
        Else
            Label41(2).BackColor = vbRed
            fxa = ((Jrk * sa) - Jrk)
        End If
        Call Lihat: Timer4.Interval = 0: Timer4.Enabled = False
    End If
ElseIf fxa <= (Jrk * (8)) Then

```



```

If fxa <= (Jrk * 3) Then
    ImgPi1(3).Visible = True: ImgPi2(3).Visible = True
    If fxa = (Jrk * 3) Then
        If RichTextBox2.Text <> Ket4(2) Then
            RichTextBox2.Text = Empty
            RichTextBox2.Text = Ket4(2)
        End If
        Label41(3).BackColor = 16776960
        Call Buka(sa)
        'isi th & thku
        LblTh.Caption = FrmSetting.CbmRngTh.Text
        LblThKu.Caption = FrmSetting.CbmRngThKu.Text
        LblPI2.Caption = FrmSetting.LblTekKel.Caption
        NCbm = (FrmSetting.CbmSPI.ListIndex + 1)
        SQL.RecordSource = "select * from " & SetData & " where Kondisi=" & NCbm
        SQL.Refresh
        If Not IsNull(SQL.Recordset(0).Value) Then RichTextBox3.Text =
SQL.Recordset(0).Value
        For i = 0 To 3
            ImgPi1(i).Visible = False: ImgPi2(i).Visible = False
        Next i
        Select Case SQL.Recordset(1).Value
            Case Is = 1, 2
                ImgPi1(1).Visible = True: ImgPi2(0).Visible = True
                fxa = ((Jrk * sa) - Jrk)
            Case Is = 10
                ImgPi1(2).Visible = True: ImgPi2(2).Visible = True
                sa = sa + 1
            Case Else
                ImgPi1(1).Visible = True: ImgPi2(1).Visible = True
                fxa = ((Jrk * sa) - Jrk)
        End Select
        FrmTrouble.Caption = SStab1.TabCaption(0) & " Tek.His = " &
LblTh.Caption & " CmHg, ThKu = " & LblThKu.Caption & " Kg/Cm 2, Tek.Keluar
= " & LblPI2.Caption & " Kg/Cm 2"
        Call Lihat: Timer4.Interval = 0: Timer4.Enabled = False
    End If
ElseIf fxa <= (Jrk * 4) Then
    ImgAmp(3).Visible = True
    If fxa = (Jrk * 4) Then
        Call Buka(sa)
        LblAmp.Caption = FrmSetting.CbmRngAmp.Text
        NCbm = (FrmSetting.CbmSAmpl.ListIndex + 1)
        SQL.RecordSource = "select * from " & SetData & " where Kondisi=" & NCbm
        SQL.Refresh
        If Not IsNull(SQL.Recordset(0).Value) Then RichTextBox3.Text =
SQL.Recordset(0).Value
        For i = 0 To 3
            ImgAmp(i).Visible = False
        Next i
    End If
End If

```



```

If SQL.Recordset(1).Value = 8 Then
    ImgAmp(2).Visible = True: sa = sa + 1
Else
    ImgAmp(1).Visible = True: fxa = ((Jrk * sa) - Jrk)
End If
FrmTrouble.Caption = SSTab1.TabCaption(0) & " Ampere = " &
LblAmp.Caption & " A"
Call Lihat: Timer4.Interval = 0: Timer4.Enabled = False
End If
ElseIf fxa <= (Jrk * 5) Then
    ImgP3(4).Visible = True
    If fxa = (Jrk * 5) Then
        Call Buka(sa)
        LblP3.Caption = Str(CDbl(LblPI2.Caption) -
CDbl(FrmSetting.CbmRngDelta(0).Text))
        NCbm = (FrmSetting.CbmSDelta(0).ListIndex + 1)
        SQL.RecordSource = "select * from " & SetData & " where Kondisi=" & NCbm
        SQL.Refresh
        If Not IsNull(SQL.Recordset(0).Value) Then RichTextBox3.Text =
SQL.Recordset(0).Value
        For i = 0 To 4
            ImgP3(i).Visible = False: ImgP3(i).Visible = False
        Next i
        Select Case SQL.Recordset(1).Value
            Case Is = 1
                ImgP3(2).Visible = True: sa = sa + 1
            Case Is = 2
                ImgP3(3).Visible = True: fxa = ((Jrk * sa) - Jrk)
            Case Is = 3
                ImgP3(1).Visible = True: fxa = ((Jrk * sa) - Jrk)
        End Select
        FrmTrouble.Caption = SSTab1.TabCaption(0) & " Delta Pi3 = " &
LblP3.Caption & " Kg/Cm 2"
        Call Lihat: Timer4.Interval = 0: Timer4.Enabled = False
    End If
ElseIf fxa <= (Jrk * 6) Then
    ImgTi1(3).Visible = True: ImgTi2(3).Visible = True
    If fxa = (Jrk * 6) Then
        Call Buka(sa)
        NCbm = (FrmSetting.CbmST1.ListIndex + 1)
        SQL.RecordSource = "select * from " & SetData & " where Kondisi=" & NCbm
        SQL.Refresh
        If Not IsNull(SQL.Recordset(0).Value) Then RichTextBox3.Text =
SQL.Recordset(0).Value
        For i = 0 To 3
            ImgTi1(i).Visible = False: ImgTi2(i).Visible = False
        Next i
        LblTi1.Caption = FrmSetting.CbmRngTI(0).Text
        Select Case SQL.Recordset(1).Value
            Case Is <= 3

```

```

    ImgTi2(0).Visible = True
    If SQL.Recordset(1).Value = 3 Then
        ImgTi1(2).Visible = True: sa = sa + 1
    Else
        ImgTi1(1).Visible = True: fxa = ((Jrk * sa) - Jrk)
    End If
    LblTi2.Caption = Empty
    Case 6, 7
        LblTi2.Caption = FrmSetting.CbmRngTi(1).Text
        ImgTi1(1).Visible = True: ImgTi2(2).Visible = True: fxa = ((Jrk * sa) - Jrk)
    Case Else
        LblTi2.Caption = FrmSetting.CbmRngTi(1).Text
        ImgTi1(1).Visible = True: ImgTi2(1).Visible = True: fxa = ((Jrk * sa) - Jrk)
    End Select
    FrmTrouble.Caption = SSTab1.TabCaption(0) & " Ti 1 = " & LblTi1.Caption
    & " oC, Ti 2 = " & LblTi2.Caption & " oC"
    Call Lihat: Timer4.Interval = 0: Timer4.Enabled = False
    End If
    ElseIf fxa <= (Jrk * 7) Then
        ImgP4(4).Visible = True
        If fxa = (Jrk * 7) Then
            Call Buka(sa)
            LblP4.Caption = Str(CDbl(LblP3.Caption) -
CDbl(FrmSetting.CbmRngDelta(1).Text))
            NCbm = (FrmSetting.CbmSDelta(1).ListIndex + 1)
            SQL.RecordSource = "select * from " & SetData & " where Kondisi=" & NCbm
            SQL.Refresh
            If Not IsNull(SQL.Recordset(0).Value) Then RichTextBox3.Text =
SQL.Recordset(0).Value
            For i = 0 To 4
                ImgP4(i).Visible = False: ImgP4(i).Visible = False
            Next i
            Select Case SQL.Recordset(1).Value
                Case Is = 1
                    ImgP4(2).Visible = True: sa = sa + 1
                Case Is = 2
                    ImgP4(3).Visible = True: fxa = ((Jrk * sa) - Jrk)
                Case Is = 3
                    ImgP4(1).Visible = True: fxa = ((Jrk * sa) - Jrk)
            End Select
            FrmTrouble.Caption = SSTab1.TabCaption(0) & " Delta Pi4 = " &
            LblP4.Caption & " Kg/Cm 2"
            Call Lihat: Timer4.Interval = 0: Timer4.Enabled = False
            sa = sa + 1
        End If
    End If
Else
    Timer4.Interval = 0: Timer4.Enabled = False
    RichTextBox3.Text = Empty: RichTextBox2.Text = Empty: Label41(3).BackColor
= vbGreen

```



```

Timer1.Interval = 200: NT = 1: Timer1.Enabled = True
End If
fxa = fxa + 1
End Sub

```

```

Private Sub Timer5_Timer()
If fxa <= (Jrk * 10) Then 'terakhir 8
    If fxa = (Jrk * 10) Then
        RichTextBox2.Text = Empty
        RichTextBox2.Text = RichTextBox2.Text & Ket5(0) & Chr(10) & Ket5(1): sa =
sa + 2
        Label51(1).BackColor = 16776960: Label51(2).BackColor = 16776960
    End If
ElseIf fxa <= (Jrk * 15) Then
    If fxa = (Jrk * 12) Then
        RichTextBox2.Text = Empty
        RichTextBox2.Text = RichTextBox2.Text & Ket5(2) & Chr(10) & Ket5(3): sa =
sa + 2
        Label51(1).BackColor = vbGreen: Label51(2).BackColor = vbGreen
        Label52(1).BackColor = 16776960: Label52(2).BackColor = 16776960
    ElseIf fxa = (Jrk * 14) Then
        Label52(1).BackColor = vbGreen: Label52(2).BackColor = vbGreen: sa = sa + 2
    End If
Else
    Timer5.Interval = 0: Timer5.Enabled = False
    RichTextBox3.Text = Empty
    Timer1.Interval = 200: NT = 1: Timer1.Enabled = True
End If
fxa = fxa + 1
End Sub

```

```

Private Sub Timer6_Timer()
If fxa <= (Jrk * 17) Then 'terakhir 15
    If fxa = (Jrk * 17) Then
        RichTextBox2.Text = Ket6(0): sa = sa + 2
        Label61(1).BackColor = 16776960
    End If
ElseIf fxa <= (Jrk * 22) Then
    If fxa = (Jrk * 19) Then
        RichTextBox2.Text = Ket6(1): sa = sa + 2
        Label61(1).BackColor = vbGreen
        Label61(2).BackColor = 16776960
    ElseIf fxa = (Jrk * 21) Then
        Label61(2).BackColor = vbGreen: sa = sa + 2
    End If
Else
    Timer6.Interval = 0: Timer6.Enabled = False
    RichTextBox3.Text = Empty
    Timer1.Interval = 200: NT = 1: Timer1.Enabled = True
End If

```



```

fxa = fxa + 1
End Sub

Private Sub Timer7_Timer()
If fxa <= (Jrk * 24) Then 'terakhir 22
    If fxa = (Jrk * 24) Then
        RichTextBox2.Text = Ket7(0)
        Label71(1).BackColor = 16776960: sa = sa + 2
    End If
ElseIf fxa <= (Jrk * 26) Then
    If fxa = (Jrk * 26) Then
        RichTextBox2.Text = Ket7(1)
        Label71(1).BackColor = vbGreen: sa = sa + 2
        Label71(2).BackColor = 16776960
    End If
ElseIf fxa <= (Jrk * 28) Then
    If fxa = (Jrk * 28) Then
        RichTextBox2.Text = Ket7(2)
        Label71(2).BackColor = vbGreen: sa = sa + 2
        Label71(3).BackColor = 16776960
    End If
ElseIf fxa <= (Jrk * 30) Then
    If fxa = (Jrk * 30) Then
        RichTextBox2.Text = Ket7(3)
        Label71(3).BackColor = vbGreen: sa = sa + 2
        Label71(4).BackColor = 16776960
    End If
ElseIf fxa <= (Jrk * 35) Then
    If fxa = (Jrk * 32) Then
        RichTextBox2.Text = Ket7(4)
        Label71(4).BackColor = vbGreen: sa = sa + 2
        Label71(5).BackColor = 16776960
    ElseIf fxa = (Jrk * 34) Then
        Label71(5).BackColor = vbGreen
    End If
Else
    Timer7.Interval = 0: Timer7.Enabled = False
    Timer1.Interval = 200: NT = 1: Timer1.Enabled = True
End If
fxa = fxa + 1
End Sub

```

```

Private Sub Timer8_Timer()
If fxa <= (Jrk * 37) Then 'terakhir 35
    If fxa = (Jrk * 37) Then
        RichTextBox2.Text = Ket8(0)
        Label81(1).BackColor = 16776960
    End If
ElseIf fxa <= (Jrk * 42) Then
    If fxa = (Jrk * 39) Then

```



```
RichTextBox2.Text = Ket7(1): sa = sa + 2
Label81(1).BackColor = vbGreen: Label81(2).BackColor = 16776960
Elseif fxa = (Jrk * 41) Then
    Label81(2).BackColor = vbGreen
End If
Else
    Timer8.Interval = 0: Timer8.Enabled = False
    Timer1.Interval = 200: NT = 1: Timer1.Enabled = True
End If
fxa = fxa + 1
End Sub
```

LIST PROGRAM UNTUK FORM SETTING

Dim Sc As Integer

Sub Bersih()

With FrmSetting

For i = 0 To .Count - 1

If Left(.Controls(i).Name, 3) = "Opt" Then

.Controls(i).Value = False

ElseIf Left(.Controls(i).Name, 3) = "Cbm" Then

If Mid(.Controls(i).Name, 4, 1) <> "S" Then

.Controls(i).Clear

Else

.Controls(i).Text = ""

End If

ElseIf Left(.Controls(i).Name, 3) = "Lbl" Then

.Controls(i).Caption = Empty

End If

Next i

End With

End Sub

Sub SyaratKeluar(ByRef Boleh As Boolean)

Dim JmlCtrl As Integer

With FrmSetting

For i = 0 To .Count - 1

If Left(.Controls(i).Name, 3) = "Cbm" Then

If Mid(.Controls(i).Name, 4, 1) = "S" Then

JmlCtrl = JmlCtrl + 1

If .Controls(i).Text <> "" Then

Sc = Sc + 1

Else

Sc = Sc - 1

End If

End If

End If

Next i

End With

JmlCtrl = JmlCtrl + 1

If Not OptRpm(0).Value And Not OptRpm(1).Value Then

Sc = Sc - 1

Else

Sc = Sc + 1

End If

MsgBox JmlCtrl & " " & Sc

If JmlCtrl <> Sc Then

MsgBox "anda teliti lagi sebelum keluar masalahnya ada inputan yg masih kosong",
vbInformation + vbOKOnly, "Hidari Error"

Boleh = False


```

Else
    Bolch = True
End If
End Sub

```

```

Sub IsiTekKel1(VarCari As Double, NCar As Integer)
If NCar = 1 Then
    SQL.RecordSource = "SELECT A_TekKel.Nil, A_TekKel.MF,
Cdbl([A_TekKel]![TekHis]) AS TekHis, " &
        "Cdbl([A_TekKel]![ThKu]) AS Thku, A_TekKel.TekKel FROM A_TekKel "
    & _
        "WHERE (((Cdbl([A_TekKel]![TekHis]))=" & VarCari & "));"
Else
    SQL.RecordSource = "SELECT A_TekKel.Nil, A_TekKel.MF,
Cdbl([A_TekKel]![TekHis]) AS TekHis, " &
        "Cdbl([A_TekKel]![ThKu]) AS Thku, A_TekKel.TekKel FROM A_TekKel "
    & _
        "WHERE (((Cdbl([A_TekKel]![TekHis]))=" & VarCari & ") AND
((Cdbl([A_TekKel]![ThKu]))=" & Cdbl(CbmRngThKu.Text) & "));"
End If
SQL.Refresh
If SQL.Recordset.RecordCount > 0 Then
    LblTekKel.Caption = SQL.Recordset(4)
End If
End Sub

```

```

Sub IsiRngTh(ByVal kds As String)
CbmRngTh.Enabled = True
CbmRngTh.BackColor = vbWhite
CbmRngTh.Clear
Select Case kds
    Case Is = "Sr"
        With CbmRngTh
            .AddItem "0", 0
            .AddItem "1", 1
            .AddItem "2", 2
            .AddItem "3", 3
            .AddItem "4", 4
            .AddItem "5", 5
            .AddItem "6", 6
            .AddItem "6.24", 7
        End With
    Case Is = "R"
        With CbmRngTh
            .AddItem "6.25", 0
            .AddItem "7", 1
            .AddItem "8", 2
            .AddItem "9", 3
            .AddItem "10", 4
            .AddItem "11", 5

```

```
.AddItem "12", 6  
.AddItem "13", 7  
.AddItem "14", 8  
.AddItem "15", 9  
.AddItem "16", 10  
.AddItem "17", 11  
.AddItem "18", 12  
.AddItem "18.74", 13
```

End With

Case Is = "N"

With CbmRngTh

```
.AddItem "18.75", 0  
.AddItem "19", 1  
.AddItem "20", 2  
.AddItem "21", 3  
.AddItem "22", 4  
.AddItem "23", 5  
.AddItem "24", 6  
.AddItem "25", 7  
.AddItem "26", 8  
.AddItem "27", 9  
.AddItem "28", 10  
.AddItem "29", 11  
.AddItem "30", 12  
.AddItem "31", 13  
.AddItem "31.24", 14
```

End With

Case Is = "T"

With CbmRngTh

```
.AddItem "31.25", 0  
.AddItem "32", 1  
.AddItem "33", 2  
.AddItem "34", 3  
.AddItem "35", 4  
.AddItem "36", 5  
.AddItem "37", 6  
.AddItem "38", 7  
.AddItem "39", 8  
.AddItem "40", 9  
.AddItem "41", 10  
.AddItem "42", 11  
.AddItem "43", 12  
.AddItem "43.74", 13
```

End With

Case Is = "St"

With CbmRngTh

```
.AddItem "43.75", 0  
.AddItem "44", 1  
.AddItem "45", 2  
.AddItem "46", 3
```

```

        .AddItem "47", 4
        .AddItem "48", 5
        .AddItem "49", 6
        .AddItem "50", 7
    End With
End Select
CbmRngTh.ListIndex = 0
End Sub

Sub IsiRngThKu(ByVal kds As String, Nil As Integer)
Dim kc, bs, i As Variant, a As Integer
Const bg As Variant = 0.1: i = 0
CbmRngThKu.Clear
CbmRngThKu.Enabled = False
CbmRngThKu.BackColor = -2147483633
If Nil = 0 Then Exit Sub
CbmRngThKu.Enabled = True
CbmRngThKu.BackColor = vbWhite
For a = 1 To Nil
    If Nil > 1 Then
        Select Case a
            Case Is = 1
                kds = "Sk"
            Case Is = 2
                kds = "K"
            Case Is = 3
                kds = "S"
            Case Is = 4
                kds = "B"
            Case Is = 5
                kds = "Sb"
        End Select
    End If
    Select Case kds
        Case Is = "Sk"
            kc = 0: bs = 0.5
        Case Is = "K"
            kc = 0.5: bs = 1.5
        Case Is = "S"
            kc = 1.5: bs = 2.5
        Case Is = "B"
            kc = 2.5: bs = 3.5
        Case Is = "Sb"
            kc = 3.5: bs = 4
    End Select
    Do While bs
        If (CDec(kc) < CDec(bs)) Then
            CbmRngThKu.AddItem kc, i
        ElseIf (CInt(kc) = 4) Then
            CbmRngThKu.AddItem kc, i
        End If
    Loop
Next a
End Sub

```



```

    CbmRngThKu.ListIndex = 0
    Exit Do
Else
    CbmRngThKu.RemoveItem (i - 1)
    CbmRngThKu.AddItem (kc - 0.1), (i - 1)
    CbmRngThKu.ListIndex = 0
    Exit Do
End If
i = i + 1
kc = kc + bg
Loop
If a = 5 Then CbmRngThKu.RemoveItem CbmRngThKu.ListCount - 1
Next a
End Sub

```

```

Sub IsiT1(ByVal kds As String, ctr As ComboBox)
Dim kc, bs, i As Variant
Const bg As Variant = 0.1: i = 0
ctr.Clear
ctr.Enabled = True
ctr.BackColor = -2147483643
Select Case kds
    Case Is = "Sk"
        kc = 25: bs = 27.5
    Case Is = "K"
        kc = 27.5: bs = 32.5
    Case Is = "N"
        kc = 32.5: bs = 37.5
    Case Is = "E"
        kc = 37.5: bs = 42.5
    Case Is = "Se"
        kc = 42.5: bs = 45
End Select
Do While bs
    If (CDec(kc) < CDec(bs)) Then
        ctr.AddItem kc, i
    ElseIf (CInt(kc) = 45) Then
        ctr.RemoveItem (i - 1)
        ctr.ListIndex = 0
        Exit Do
    Else
        If (CDec(kc) = CDec(bs)) Then
            ctr.RemoveItem (i - 1)
            ctr.AddItem (kc - 0.1), (i - 1)
            ctr.ListIndex = 0
            Exit Do
        End If
    End If
    i = i + 1
    kc = kc + bg

```

```
Loop
End Sub
```

```
Sub IsiRngAmp(ByVal kds As String)
Dim kc, bs, i As Variant
Const bg As Variant = 0.1: i = 0
CbmRngAmp.Clear
Select Case kds
Case Is = "Sk"
kc = 7: bs = 7.5
Case Is = "K"
kc = 7.5: bs = 8.5
Case Is = "N"
kc = 8.5: bs = 9.5
Case Is = "B"
kc = 9.5: bs = 10.5
Case Is = "Sb"
kc = 10.5: bs = 11
End Select
Do While bs
If (CDec(kc) < CDec(bs)) Then
CbmRngAmp.AddItem kc, i
ElseIf (CInt(kc) = 11) Then
CbmRngAmp.RemoveItem (i - 1)
CbmRngAmp.ListIndex = 0
Exit Do
Else
If (CDec(kc) = CDec(bs)) Then
CbmRngAmp.RemoveItem (i - 1)
CbmRngAmp.AddItem (kc - 0.1), (i - 1)
CbmRngAmp.ListIndex = 0
Exit Do
End If
End If
i = i + 1
kc = kc + bg
Loop
End Sub
```

```
Private Sub CbmRngAmp_Click()
SQL.RecordSource = "SELECT CDBl([A_PanBG]![VarAmp]) AS VarAmp,
A_PanBG.Pan, A_PanBG.Bg2 " & _
"From A_PanBG " & _
"WHERE (((CDBl([A_PanBG]![VarAmp]))= " & CbmRngAmp.Text & "));"
SQL.Refresh
LblBG2.Caption = SQL.Recordset(2)
LblPan.Caption = SQL.Recordset(1)
End Sub
```

```
Private Sub CbmRngAmp_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
    CbmST1.SetFocus
Else
    KeyAscii = 0
End If
End Sub
```

```
Private Sub CbmRngTh_Click()
If Not CbmRngThKu.Enabled Then
    Call IsiTekKel1(CDbl(CbmRngTh.Text), 1)
End If
End Sub
```

```
Private Sub CbmRngThKu_Click()
If Not CbmRngTh.Enabled Then
    LblTekKel.Caption = 0
Else
    Call IsiTekKel1(CDbl(CbmRngTh.Text), 2)
End If
End Sub
```

```
Private Sub CbmRngTI_KeyPress(Index As Integer, KeyAscii As Integer)
If KeyAscii = 13 Then
    Select Case Index
        Case Is = 0
            If CbmRngTI(1).Enabled Then
                CbmRngTI(1).SetFocus
            Else
                CbmSDelta(0).SetFocus
            End If
        Case Is = 1
            CbmSDelta(0).SetFocus
    End Select
Else
    KeyAscii = 0
End If
End Sub
```

```
Private Sub CbmSAmp1_Click()
Select Case CbmSAmp1.ListIndex
    Case Is = 0, 5 '1,6
        Call IsiRngAmp("Sk")
    Case Is = 1, 6 '2,7
        Call IsiRngAmp("K")
    Case Is = 2, 7 '3,8
        Call IsiRngAmp("N")
    Case Is = 3, 8 '4,9
        Call IsiRngAmp("B")
    Case Is = 4, 9 '5,10
        Call IsiRngAmp("Sb")
End Select
```


End Select

End Sub

Private Sub CbmSDelta_Click(Index As Integer)

CbmRngDelta(Index).Clear

Select Case CbmSDelta(Index).ListIndex

Case Is = 0

CbmRngDelta(Index).AddItem "0", 0 '1

CbmRngDelta(Index).AddItem "0.05", 1 '2

CbmRngDelta(Index).AddItem "0.1", 2 '3

CbmRngDelta(Index).AddItem "0.124", 3 '4

Case Is = 1

CbmRngDelta(Index).AddItem "0.125", 0 '1

CbmRngDelta(Index).AddItem "0.15", 1 '2

CbmRngDelta(Index).AddItem "0.2", 2 '3

CbmRngDelta(Index).AddItem "0.25", 3 '4

CbmRngDelta(Index).AddItem "0.3", 4 '5

CbmRngDelta(Index).AddItem "0.35", 5 '6

CbmRngDelta(Index).AddItem "0.374", 6 '7

Case Is = 2

CbmRngDelta(Index).AddItem "0.375", 0 '1

CbmRngDelta(Index).AddItem "0.4", 1 '2

CbmRngDelta(Index).AddItem "0.45", 2 '3

CbmRngDelta(Index).AddItem "0.5", 3 '4

End Select

CbmRngDelta(Index).ListIndex = 0

End Sub

Private Sub CbmSDelta_KeyPress(Index As Integer, KeyAscii As Integer)

If KeyAscii = 13 Then

CbmRngDelta(Index).SetFocus

Else

KeyAscii = 0

End If

End Sub

Private Sub CbmRngDelta_KeyPress(Index As Integer, KeyAscii As Integer)

If KeyAscii = 13 Then

Select Case Index

Case Is = 0

CbmSDelta(Index + 1).SetFocus

Case Is = 1

CbmSumpTank.SetFocus

End Select

Else

KeyAscii = 0

End If

End Sub

Private Sub CbmRngSumpTank_KeyPress(KeyAscii As Integer)

```
If KeyAscii = 13 Then
    OptRpm(0).SetFocus
Else
    KeyAscii = 0
End If
End Sub
```

```
Private Sub CbmRngTh_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    If CbmRngThKu.Enabled Then
        CbmRngThKu.SetFocus
    Else
        CbmSAmp1.SetFocus
    End If
Else
    KeyAscii = 0
End If
End Sub
```

```
Private Sub CbmRngThKu_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    CbmSAmp1.SetFocus
Else
    KeyAscii = 0
End If
End Sub
```

```
Private Sub CbmSAmp1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    CbmRngAmp.SetFocus
Else
    KeyAscii = 0
End If
End Sub
```

```
Private Sub CbmSP1_Click()
Select Case CbmSP1.ListIndex
Case Is = 0, 12 '1'13
    CbmRngTh.Clear
    CbmRngTh.BackColor = -2147483633
    CbmRngTh.Enabled = False
    Call IsiRngThKu("Sk", 5)
    CbmRngThKu.ListIndex = 1
    CbmRngThKu.RemoveItem 0
Case Is = 1 '2
    CbmRngTh.Enabled = True
    CbmRngTh.BackColor = vbWhite
    CbmRngTh.Clear
```

```

    CbmRngTh.AddItem "0", 0
    CbmRngTh.ListIndex = 0
    CbmRngThKu.Clear
    CbmRngThKu.AddItem "0", 0
    CbmRngThKu.ListIndex = 0
    LblTekKel = 0
Case Is = 2 '3
    Call IsiRngTh("Sr")
    Call IsiRngThKu("Sb", 1)
Case Is = 3 '4
    Call IsiRngTh("Sr")
    Call IsiRngThKu("B", 1)
Case Is = 4 '5
    Call IsiRngTh("Sr")
    Call IsiRngThKu("S", 1)
Case Is = 5 '6
    Call IsiRngTh("Sr")
    Call IsiRngThKu("K", 1)
Case Is = 6 '7
    Call IsiRngTh("Sr")
    Call IsiRngThKu("Sk", 1)
Case Is = 7, 8 '8'9
    Call IsiRngTh("R")
    Call IsiRngThKu("Sk", 1)
Case Is = 9 '10
    Call IsiRngTh("N")
    Call IsiRngThKu("Hl", 0)
    Call IsiTekKel1(CDbl(CbmRngTh.Text), 1)
Case Is = 10 '11
    Call IsiRngTh("T")
    Call IsiRngThKu("Hl", 0)
Case Is = 11 '12
    Call IsiRngTh("St")
    Call IsiRngThKu("Hl", 0)
Case Is = 13 '14
    Call IsiRngTh("Sr")
    Call IsiRngThKu("Hl", 0)
Case Is = 14 '15
    Call IsiRngTh("R")
    Call IsiRngThKu("Hl", 0)
End Select

End Sub

Private Sub CbmSP1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    If CbmRngTh.Enabled Then
        CbmRngTh.SetFocus
    Else
        CbmRngThKu.SetFocus
    End If
End If

```



```

End If
Else
    KeyAscii = 0
End If
End Sub

Private Sub CbmST1_Click()
Select Case CbmST1.ListIndex
    Case Is = 0 '1
        Call IsiT1("Sk", CbmRngTI(0))
        CbmRngTI(1).Clear
        CbmRngTI(1).BackColor = -2147483633
        CbmRngTI(1).Enabled = False
    Case Is = 1 '2
        Call IsiT1("K", CbmRngTI(0))
        CbmRngTI(1).Clear
        CbmRngTI(1).BackColor = -2147483633
        CbmRngTI(1).Enabled = False
    Case Is = 2 '3
        Call IsiT1("N", CbmRngTI(0))
        CbmRngTI(1).Clear
        CbmRngTI(1).BackColor = -2147483633
        CbmRngTI(1).Enabled = False
    Case Is = 3 '4
        Call IsiT1("E", CbmRngTI(0))
        Call IsiT1("E", CbmRngTI(1))
    Case Is = 4 '5
        Call IsiT1("Se", CbmRngTI(0))
        Call IsiT1("Se", CbmRngTI(1))
    Case Is = 5 '6
        Call IsiT1("E", CbmRngTI(0))
        Call IsiT1("N", CbmRngTI(1))
    Case Is = 6 '7
        Call IsiT1("Sk", CbmRngTI(0))
        Call IsiT1("N", CbmRngTI(1))
    Case Is = 7 '8
        Call IsiT1("Sk", CbmRngTI(0))
        Call IsiT1("E", CbmRngTI(1))
End Select
End Sub

```

```

Private Sub CbmST1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    CbmRngTI(0).SetFocus
Else
    KeyAscii = 0
End If
End Sub

```

```

Private Sub CbmSumpTank_Click()

```

```

Dim kc, bs, i As Integer
Const bg As Integer = 50: i = 0
CbmRngSumpTank.Clear
Select Case CbmSumpTank.ListIndex
    Case Is = 0
        kc = 200: bs = 400
        AVol = "Srd"
    Case Is = 1
        kc = 400: bs = 800
        AVol = "R"
    Case Is = 2
        kc = 800: bs = 1200
        AVol = "S"
    Case Is = 3
        kc = 1200: bs = 1600
        AVol = "B"
    Case Is = 4
        kc = 1600: bs = 1800
        AVol = "Sb"
End Select
Do While bs
    CbmRngSumpTank.AddItem kc, i
    i = i + 1
    kc = kc + bg
    If kc = 1800 Then
        CbmRngSumpTank.AddItem kc, i
        CbmRngSumpTank.ListIndex = 0
        Exit Do
    Else
        If (kc = bs) Then
            CbmRngSumpTank.AddItem kc - 1, i
            CbmRngSumpTank.ListIndex = 0
            Exit Do
        End If
    End If
Loop
End Sub

Private Sub CbmSumpTank_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    CbmRngSumpTank.SetFocus
Else
    KeyAscii = 0
End If
End Sub

```

```

Private Sub Check1_Click()
If Check1.Value = vbUnchecked Then
    SAut = True

```



```

Else
    SAut = False
End If
End Sub

```

```

Private Sub Check2_Click()
If Check2.Value = vbChecked Then
    CbmSumpTank.ListIndex = 2: CbmSP1.ListIndex = 9
    CbmSAmp1.ListIndex = 7: CbmST1.ListIndex = 2
    CbmSDelta(0).ListIndex = 0: CbmSDelta(1).ListIndex = 0
    Check3.Value = vbChecked: OptRpm(0).Value = True
Else
    Call Bersih
End If
End Sub

```

```

Private Sub Check3_Click()
For i = 0 To 1
    OptRpm(i).Value = False
Next i
If Check3.Value = vbChecked Then
    Label1A(30).Enabled = True: Label1A(31).Enabled = True
    OptRpm(0).Enabled = True: OptRpm(1).Enabled = True
Else
    Label1A(30).Enabled = False: Label1A(31).Enabled = False
    OptRpm(0).Enabled = False: OptRpm(1).Value = True
End If
End Sub

```

```

Private Sub Form_Load()
SQL.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
App.Path & "\VBaru.mdb;Persist Security Info=False"
SQL.CommandType = adCmdText
CbmSumpTank.AddItem "Sangat Rendah", 0 'Srd
CbmSumpTank.AddItem "Rendah", 1 'Rd
CbmSumpTank.AddItem "Sedang", 2 'S
CbmSumpTank.AddItem "Banyak", 3 'B
CbmSumpTank.AddItem "Sangat Banyak", 4 'Sb
PI 1 & PI 2
CbmSP1.AddItem "1). TH < 0, THKU = 1/4, TK = 0", 0 '1
CbmSP1.AddItem "2). TH = 0, THKU = 0, TK = SK", 1 '2
CbmSP1.AddItem "3). TH = SR, THKU = SB, TK = SK", 2 '3
CbmSP1.AddItem "4). TH = SR, THKU = B, TK = SK", 3 '4
CbmSP1.AddItem "5). TH = SR, THKU = S, TK = K", 4 '5
CbmSP1.AddItem "6). TH = SR, THKU = K, TK = K", 5 '6
CbmSP1.AddItem "7). TH = SR, THKU = SK, TK = K", 6 '7
CbmSP1.AddItem "8). TH = R, THKU = SK, TK = K", 7 '8
CbmSP1.AddItem "9). TH = R, THKU = SK, TK = K", 8 '9
CbmSP1.AddItem "10). TH = N," & Space(27) & "TK = N", 9 '10
CbmSP1.AddItem "11). TH = T," & Space(27) & "TK = B", 10 '11

```



```

CbmSP1.AddItem "12).TH = ST," & Space(25) & "TK = SB", 11 '12
CbmSP1.AddItem "13).TH < 0,   THKU = 1/4, TK = 0", 12 '13
CbmSP1.AddItem "14).TH = SR," & Space(25) & "TK = SK", 13 '14
CbmSP1.AddItem "15).TH = R," & Space(27) & "TK = SB", 14 '15
'Ampere
CbmSAmpl.AddItem "1). Am = SK, BG2 = SR", 0 ' 1
CbmSAmpl.AddItem "2). Am = K,  BG2 = R", 1 ' 2
CbmSAmpl.AddItem "3). Am = N,  BG2 = S", 2 ' 3
CbmSAmpl.AddItem "4). Am = B,  BG2 = K", 3 ' 4
CbmSAmpl.AddItem "5). Am = SB, BG2 = SK", 4 ' 5
CbmSAmpl.AddItem "6). Am = SK, Pan = SR", 5 ' 6
CbmSAmpl.AddItem "7). Am = K,  Pan = R", 6 ' 7
CbmSAmpl.AddItem "8). Am = N,  Pan = N", 7 ' 8
CbmSAmpl.AddItem "9). Am = B,  Pan = T", 8 ' 9
CbmSAmpl.AddItem "10).Am = SB, Pan = ST", 9 ' 10
'TI 1 & TI 2
CbmST1.AddItem "1). TI 1 = Sr", 0 '1
CbmST1.AddItem "2). TI 1 = R", 1 '2
CbmST1.AddItem "3). TI 1 = N", 2 '3
CbmST1.AddItem "4). TI 1 = T, TI 2 = T", 3 '4
CbmST1.AddItem "5). TI 1 = St, TI 2 = St", 4 '5
CbmST1.AddItem "6). TI 1 = T, TI 2 = N", 5 '6
CbmST1.AddItem "7). TI 1 = St, TI 2 = N", 6 '7
CbmST1.AddItem "8). TI 1 = St, TI 2 = T", 7 '8
'PI 3 & PI 4
For i = 0 To 1
    CbmSDelta(i).AddItem "Kecil", 0 '1
    CbmSDelta(i).AddItem "Sedang", 1 '2
    CbmSDelta(i).AddItem "Besar", 2 '3
Next i

End Sub

Private Sub Form_Unload(Cancel As Integer)
Dim Boleh As Boolean
Sc = 0
Call SyaratKeluar(Boleh)
If Boleh Then
    Jln = True
    If Strb Then
        SSet = True
    Else
        SSet = False
    End If
    Cancel = 1: Me.Visible = False
    MnuUtama.Show
Else
    Cancel = -1
End If
End Sub

```

```
Private Sub OptRpm_Click(Index As Integer)
    CbmSP1.SetFocus
End Sub
```

```
Private Sub OptRpm_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then CbmSP1.SetFocus
End Sub
```

```
Private Sub SStab1_DblClick()
```

```
End Sub
```

LIST PROGRAM UNTUK FORM TROUBLE

Dim fi, fx, fa, ai As Integer

Sub Jalan()

Dim Sama As Boolean: Sama = True

Do While Sama

 SQL.Recordset.MoveNext

 If SQL.Recordset.EOF = True Then

 CmdNext.Enabled = False

 SQL.Recordset.MoveLast

 Exit Do

 End If

 If SQL.Recordset(3) = Text1.Text Then

 RichTextBox1.Text = RichTextBox1.Text + Chr(10)

 RichTextBox1.Text = RichTextBox1.Text & (SQL.Recordset(4))

 Else

 SQL.Recordset.MovePrevious

 Sama = False

 End If

Loop

End Sub

Private Sub CmdBack_Click()

Dim CekAtas, CekBawah As String

Dim i, a As Integer: i = 1

CmdNext.Enabled = True

CekAtas = Left(SQL.Recordset(4), 1)

SQL.Recordset.MovePrevious

CekBawah = Left(SQL.Recordset(4), 1)

If (CekAtas = "c" And CekBawah = "b") Or (CekAtas = "d" And CekBawah = "c") _

 Or (CekAtas = "e" And CekBawah = "d") Then

 SQL.Recordset.MoveNext

 SQL.Recordset.MoveNext

 SQL.Recordset.MovePrevious

End If

Text1.Text = SQL.Recordset(3)

Dim Sama As Boolean: Sama = True

Do While Sama

 SQL.Recordset.MovePrevious

 If SQL.Recordset.BOF = True Then

 SQL.Recordset.MoveFirst

 RichTextBox1.Text = SQL.Recordset(4)

 CmdBack.Enabled = False

 Exit Do

 End If

 If SQL.Recordset(3) = Text1.Text Then

 i = i + 1

 Else


```

    SQL.Recordset.MoveNext
    RichTextBox1.Text = SQL.Recordset(4)
    Sama = False
End If
Loop
If i > 1 Then
    Sama = True
    a = i
    Do While Sama
        i = i - 1
        SQL.Recordset.MoveNext
        Text1.Text = SQL.Recordset(3)
        RichTextBox1.Text = RichTextBox1.Text + Chr(10)
        RichTextBox1.Text = RichTextBox1.Text & SQL.Recordset(4)
        If i = 1 Then Exit Do
    Loop
    For i = 1 To a - 1
        SQL.Recordset.MovePrevious
    Next i
End If

End Sub

```

```

Private Sub CmdExit_Click()
    CmdNext.Enabled = False: CmdBack.Enabled = False
    OptSct.Enabled = True: OptSys.Enabled = True: OptMch.Enabled = True
    Timer1.Interval = 0: Timer1.Enabled = False: OptSys.BackColor = -2147483648#
    Timer2.Interval = 0: Timer2.Enabled = False: OptSct.BackColor = -2147483648#
    Timer3.Interval = 0: Timer3.Enabled = False: OptMch.BackColor = -2147483648#
    Text1.Text = Empty: RichTextBox1.Text = Empty
    OptSct.Value = False: OptSys.Value = False: OptMch.Value = False
    MnuUtama.Timer4.Interval = 200: NT = 4
    MnuUtama.Timer4.Enabled = True
    Me.Hide: SSet = True: Strb = True
End Sub

```

```

Private Sub CmdNext_Click()
    Dim CekRelation As String
    If SQL.Recordset.EOF = True Then
        CmdNext.Enabled = False
        Exit Sub
    End If
    If SQL.Recordset.BOF = True Then
        SQL.Recordset.MoveFirst
    End If
    CmdBack.Enabled = True
    CekRelation = Left((SQL.Recordset(4)), 1)
    If CekRelation = "a" Then
        SQL.Recordset.MovePrevious
    End If

```

```

SQL.Recordset.MoveNext
Text1.Text = SQL.Recordset(3)
RichTextBox1.Text = (SQL.Recordset(4))
Call Jalan
End Sub

```

```

Private Sub Form_Activate()
fi = 0: fx = 0: fa = 0: ai = 0
If SAut Then
    Timer4.Interval = 200
    Timer4.Enabled = True
Else
    Timer4.Interval = 0
    Timer4.Enabled = False
End If
End Sub

```

```

Private Sub Form_Load()
SQL.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
App.Path & "\VBaru.mdb;Persist Security Info=False"
SQL.CommandType = adCmdText
SQL.RecordSource = "select * from " & SetData
SQL.Refresh
End Sub

```

```

Private Sub OptMch_Click()
If OptMch.Value = True Then
    SQL.RecordSource = "select * from " & SetData & _
        " where Kondisi=" & NCbm & " And Troubles=" & _
        Left(OptMch.Caption, 1) & _
        Format(Right(OptMch.Caption, Len(OptMch.Caption) - 1), "<") & ""
    SQL.Refresh
    If SQL.Recordset.RecordCount = 0 Then
        Text1.Text = Empty
        RichTextBox1.Text = Empty
        CmdNext.Enabled = False
        CmdBack.Enabled = False
        Exit Sub
    Else
        SQL.Recordset.MoveFirst
        CmdNext.Enabled = True
        CmdBack.Enabled = False
    End If
    Text1.Text = SQL.Recordset(3)
    RichTextBox1.Text = (SQL.Recordset(4))
    Call Jalan
End If
End Sub

```

```

Private Sub OptSet_Click()

```

```

If OptSct.Value = True Then
    SQL.RecordSource = "select * from " & SetData & _
        " where Kondisi=" & NCbm & " And Troubles=" & _
        Left(OptSct.Caption, 1) & _
        Format(Right(OptSct.Caption, Len(OptSct.Caption) - 1), "<") & ""
    SQL.Refresh
    If SQL.Recordset.RecordCount = 0 Then
        Text1.Text = Empty
        RichTextBox1.Text = Empty
        CmdNext.Enabled = False
        CmdBack.Enabled = False
        Exit Sub
    Else
        SQL.Recordset.MoveFirst
        CmdNext.Enabled = True
        CmdBack.Enabled = False
    End If
    Text1.Text = SQL.Recordset(3)
    RichTextBox1.Text = (SQL.Recordset(4))
    Call Jalan
End If
End Sub

```

```

Private Sub OptSys_Click()
If OptSys.Value = True Then
    SQL.RecordSource = "select * from " & SetData & _
        " where Kondisi=" & NCbm & " And Troubles=" & _
        Left(OptSys.Caption, 1) & _
        Format(Right(OptSys.Caption, Len(OptSys.Caption) - 1), "<") & ""
    SQL.Refresh
    If SQL.Recordset.RecordCount = 0 Then
        Text1.Text = Empty
        RichTextBox1.Text = Empty
        CmdNext.Enabled = False
        CmdBack.Enabled = False
        Exit Sub
    Else
        SQL.Recordset.MoveFirst
        CmdNext.Enabled = True
        CmdBack.Enabled = False
    End If
    Text1.Text = SQL.Recordset(3)
    RichTextBox1.Text = (SQL.Recordset(4))
    Call Jalan
End If
End Sub

```

```

Private Sub Timer1_Timer()
If fx < 5 Then

```



```
    OptSct.BackColor = 255
ElseIf fx < 6 Then
    OptSct.BackColor = -2147483648#
Else
    fx = 0
End If
fx = fx + 1
End Sub
```

```
Private Sub Timer2_Timer()
If fa < 5 Then
    OptSys.BackColor = 255
ElseIf fa < 6 Then
    OptSys.BackColor = -2147483648#
Else
    fa = 0
End If
fa = fa + 1
End Sub
```

```
Private Sub Timer3_Timer()
If fi < 5 Then
    OptMch.BackColor = 255
ElseIf fi < 6 Then
    OptMch.BackColor = -2147483648#
Else
    fi = 0
End If
fi = fi + 1
End Sub
```

```
Private Sub Timer4_Timer()
If ai = 10 Then
    CmdExit_Click
End If
ai = ai + 1
End Sub
```

General Setting [X]

Pompa Oli CFW CSW General

| | | |
|--|---|--|
| Volume Sump tank [] | Seting PI 1 dan PI 2 [] | Seting Ampere [] |
| Range Volume Sump tank [] Liter | Range Tekan Hisap [] cmHg | Range Ampere [] A |
| Rpm <input checked="" type="checkbox"/> On <input type="radio"/> ≤ 1150 <input type="radio"/> 0 | Range Tekanan Hisap Kemasukan Udara [] Kg / cm ² | Bunyi & Get 2 [] % |
| | Tekan Keluar [] Kg / cm ² | Panas [] °C |
| Seting TI 1 dan TI 2 [] | Seting PI 3 [] | Seting PI 4 [] |
| Range TI 1 [] °C | Range PI 3 [] Kg / cm ² | Range PI 4 [] Kg / cm ² |
| Range TI 2 [] °C | | |

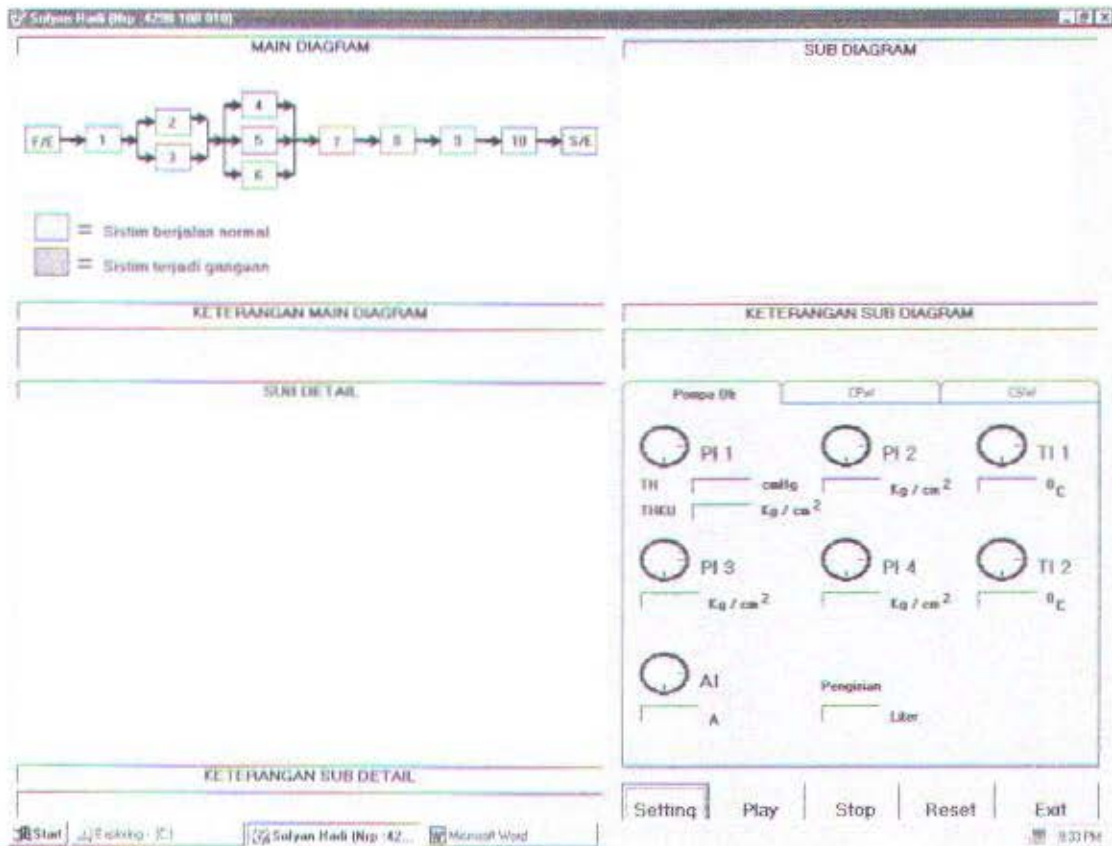
Gambar 1. Tampilan awal penyettingan.

Global

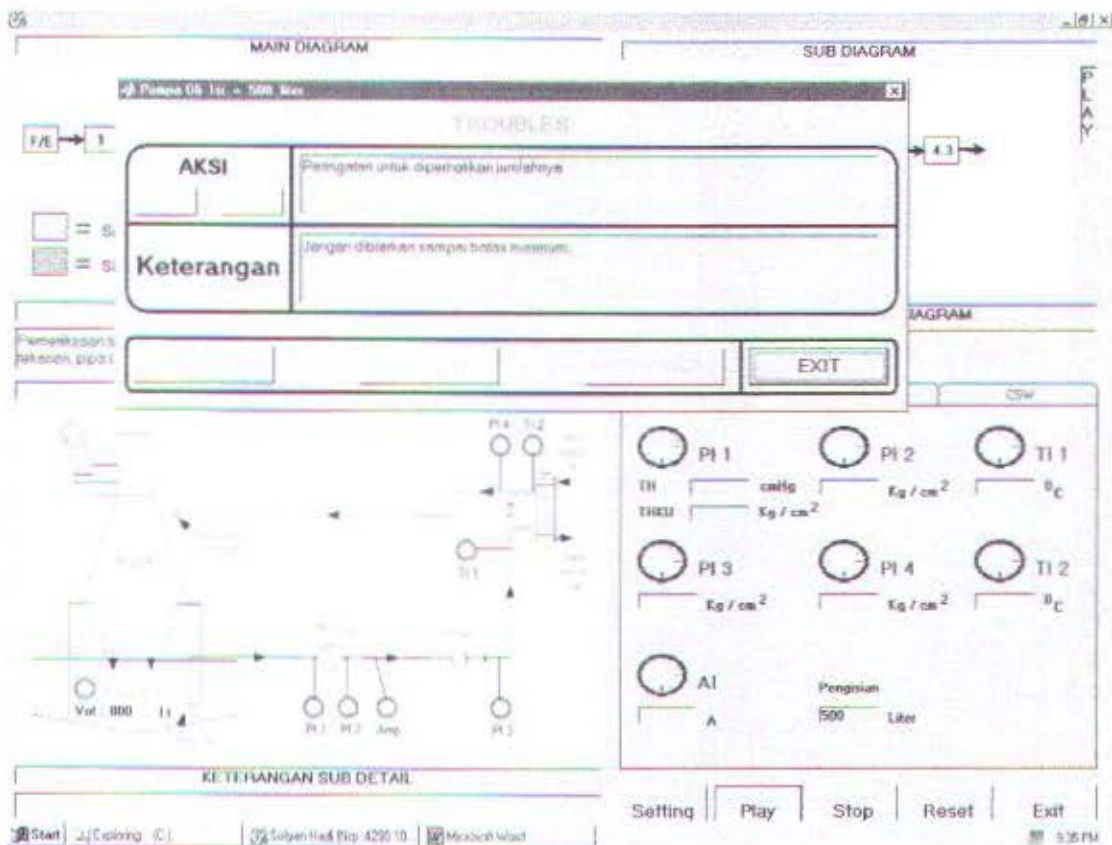
☒ Trouble Manual Close

☒ Pompa Oli Normal

Gambar 2. Tampilan penyettingan global.



Gambar 3. Tampilan awal program.

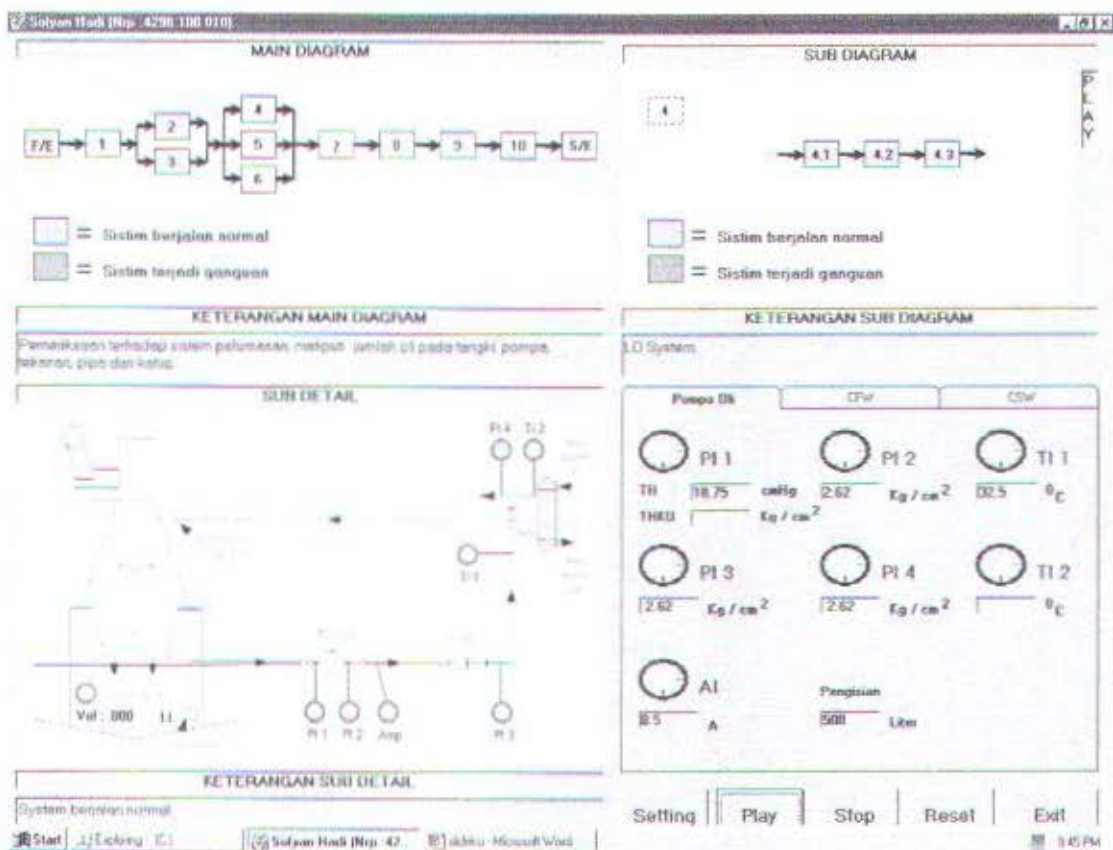


Gambar 4. Tampilan jalannya program untuk volume sump tank saat normal.

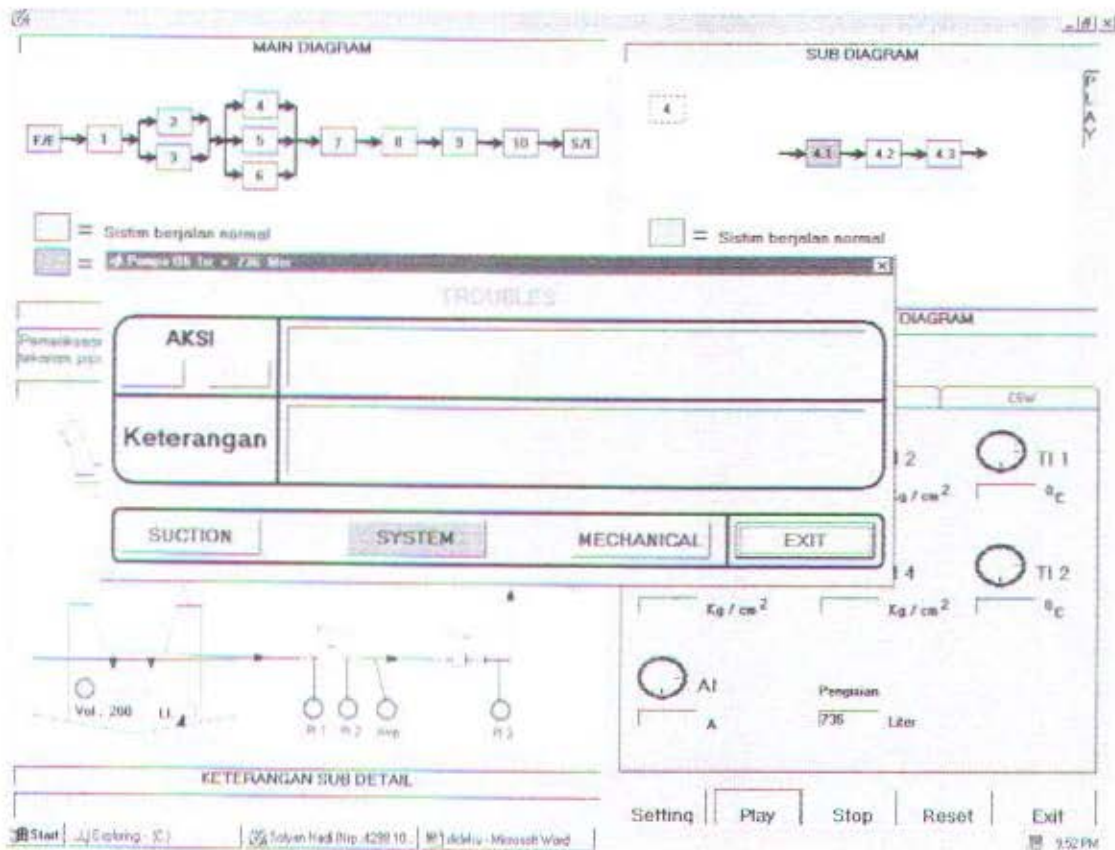
General Setting

| Pompa Oli | CFW | CSW | General |
|---|---|--------------------------------------|---------|
| Volume Sumptank Sedang | Seting PI 1 dan PI 2 10) TH = N, TK = N | Seting Ampere 8) Am = N, Pan = N | |
| Range Volume Sumptank 800 Liter | Range Tekan Hisap 18.75 cmHg | Range Ampere 8.5 A | |
| Rpm <input checked="" type="checkbox"/> On <input checked="" type="radio"/> ≤ 1150 <input type="radio"/> 0 | Range Tekanan Hisap Kemasukan Udara Kg / cm ² | Bunyi & Get 2 37.5 z | |
| | Tekan Keluar 2.62 Kg / cm ² | Panas 30 0C | |
| Seting TI 1 dan TI 2 3) TI 1 = N | Seting PI 3 Kecil | Seting PI 4 Kecil | |
| Range TI 1 32.5 0C | Range PI 3 0 Kg / cm ² | Range PI 4 0 Kg / cm ² | |
| Range TI 2 0C | | | |

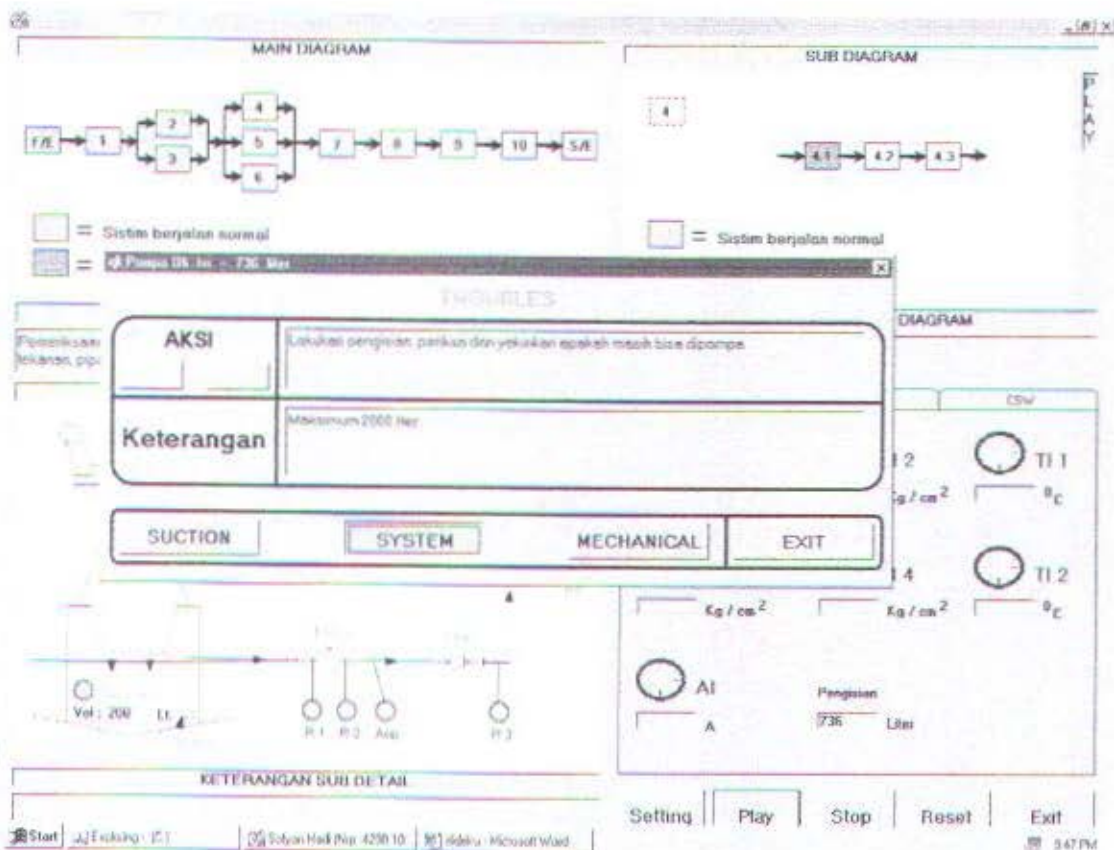
Gambar 5. Tampilan untuk sistem pelumas saat kondisi normal.



Gambar 6. Tampilan jalannya program untuk sistem pelumas saat norma



Gambar 7. Tampilan awal jalannya program untuk volume sump tank saat terjadi trouble.



Gambar 8. Tampilan jalannya program untuk volume sump tank saat terjadi trouble.